

phasing plan 1

Below is a **complete, step-by-step phasing plan** from “works but not good” → **launch**, with TestFlight at the center.

Phase 0 – Stabilize & Instrument (You + very small group, ~1–2 weeks)

Goal: Make the current app *safe to test* and observable. No new features yet.

0.1. Clean build + TestFlight baseline

1. Fix any **crash-on-launch** or obvious showstoppers.
2. Integrate:
 - Crash reporting (e.g. Xcode + any 3rd party if you use one)
 - Analytics events via PostHog (or your chosen stack).
3. Cut a **new TestFlight build**:
 - Version & build numbers correct.
 - Internal group only (you, 2–3 trusted collaborators).
4. Set up **TestFlight groups**:
 - Internal Dev (up to 25 people)
 - Friends & Family
 - Power Testers (later)
 - Public Beta (later)

0.2. Instrument the three critical flows

For this build, instrument these events:

1. **Email load**

- email_list_loaded
- email_opened

2. Summaries

- summary_requested
- summary_shown
- summary_edited

3. Actions (even if rough)

- action_suggested
- action_selected
- action_completed
- action_failed

Exit Criteria for Phase 0

- You can:
 - Install via TestFlight on at least 3 devices.
 - See crash reports and basic analytics.
 - Open 20+ real emails without crashing.
- No blocker bugs that stop basic triage.

Phase 1 – Internal Alpha (0–20 users, ~2 weeks)

Goal: Fix “it’s not good” at the *core*: reading + summarizing + not feeling broken.

1.1. Prioritize the “feel” issues

While only internal people are using it:

1. Stability pass

- Fix top 5 crash/bug patterns from Phase 0.
- Address any “I literally can’t use it” bugs.

2. Core UX pass

Focus only on:

- Swipe feeling janky / laggy.
- Email rendering issues (fonts, layout broken, scrolling).
- Modals that don’t dismiss or trap the user.

3. Summarizer v1

- Wire the actual summarization call (or stub with a real model).
- Add a simple “Was this summary helpful?” prompt (Yes/No).
- Log:
 - summary_feedback = helpful/not_helpful
 - summary_edited boolean.

1.2. Run a structured internal test

1. Give 5–10 internal testers a “script”:

- Triaging 25 real emails.
- Use summary on 10.
- Try 3 actions (even if rough).

2. Have them **record friction**:

- “I don’t trust the summary because...”
- “This action did nothing / was confusing.”
- “This crashed / locked up.”

Exit Criteria for Phase 1

- Crash-free sessions for at least 10 internal users doing 20+ emails each.
 - Summaries:
 - $\geq 70\%$ marked “helpful” by internal testers.
 - You have a **ranked list** of top 10 “this feels bad” issues.
-

Phase 2 – Friends & Family Beta (20–100 users, ~2–3 weeks)

Goal: Make the **core reading + summary experience genuinely good** for a small real-world cohort. Still de-prioritize the full action set.

2.1. Quality pass on summaries + reading

1. **Summary improvements:**
 - Tighten prompt to enforce:
 - 2–3 sentences, $\leq \sim 280$ chars.
 - No hallucinated details.
 - “If I’m not sure, I’ll say I’m not sure” behavior.
 - Add a quick “fix this” button that:
 - Lets user request a new summary or edit.
 - Logs when they do.

2. **Reading comfort:**

- Fix any font/line spacing issues.
- Improve scroll performance.
- Hide obviously broken HTML elements.

3. **Action strategy for this phase:**

- Pick **3–4 actions** to be “Phase 2 core”:
 - e.g. RSVP / Add to Calendar, Track Package, Add Reminder, Save Email.
- Everything else: either hidden or marked as “Coming soon.”

2.2. Friends & family rollout plan

1. Invite 20–100 people you trust (not strangers yet):

- Use 1–2 emails with:
 - What’s in, what’s rough, what you want feedback on.
- Give them a **short survey link** (Typeform, etc.).

2. Ask them to:

- Use Zero for **3–5 days**.
- Try summary on **10+ messages**.
- Use at least 5 actions (even if just your 3–4 core ones).

3. Capture metrics:

- Crashes per user.
- Summary helpful ratio.
- How often actions succeed vs fail.

- “Would you keep using this?” (1–5).

Exit Criteria for Phase 2

- crash rate **very low** (< 1 crash per 100 sessions).
 - Summary:
 - $\geq 80\%$ “helpful” votes.
 - Edit rate trending under ~40% (we’ll keep pushing this down).
 - At least 2 actions feel solid (users can explain what they did and why).
-

Phase 3 – Guided External Beta (100–500 users, ~3–4 weeks)

Goal: Lock core product promise:

“It helps me **understand and clear my inbox faster**, with a couple of magical actions.”

3.1. Expand to 500, but keep guardrails

1. Open TestFlight to a **waitlist** or landing page signups.
2. Admit users in waves:
 - Wave 1: 50–100
 - Wave 2: +100–200
 - Wave 3: up to 500
3. For each wave:
 - Email them a “**Day 1–3 get started**” guide:
 - How summaries work.

- Which actions are actually ready.
- What kind of feedback is most helpful.

3.2. Product changes in this phase

1. Summary & classifier hardening

- Run **manual review**:
 - Sample 100–200 summary outputs.
 - Mark each as: “Good / Meh / Bad”.
- Tune prompts / model choices to:
 - Push “Bad” below 5–10%.
 - “Good” ideally > 70%.

2. Top 10 actions – but in tiers

Break your 10 target actions into **tiers**:

- **Tier 1 (must be great now)**
e.g. RSVP/Add to Calendar, Track Package, Add Reminder.
- **Tier 2 (good enough, can improve)**
e.g. Pay invoice, Save contact, File/snooze email.
- **Tier 3 (nice, but optional)**
e.g. more niche compound flows, admin/training stuff.

In this phase:

- Make **Tier 1 rock solid** (high success, low confusion).
- Tier 2 actions: keep in but mark as “Beta” in UI if needed.

- Tier 3: consider hiding or disabling to avoid noise.

3. Onboarding & empty states

- Add a simple **first-run walkthrough**:
 - “Zero helps you...”
 - “Try a summary on a long email.”
 - “Try this one action.”
- Improve states like:
 - “No emails to show.”
 - “We had trouble summarizing this, here’s what to do instead.”

3.3. Feedback loop with power testers

Create a **Power Testers** TestFlight group:

- Invite people who:
 - Use app daily,
 - Send thoughtful feedback.
- Give them:
 - A shared doc or Notion page.
 - Occasional short surveys.
 - Access to new features **1 build earlier** than general beta.

Exit Criteria for Phase 3

- At least 100–200 external users **using it weekly**.
- Summary:

- ≥ 85% helpful.
 - Edit rate trending < 30%.
 - Tier 1 actions:
 - ≥ 95–98% success (no broken flows).
 - Users can describe the value in their own words.
 - NPS / satisfaction:
 - ≥ 4.0 / 5 from external beta users.
-

Phase 4 – Public Beta / Soft Launch (500–2,000+ users, ~4–6 weeks)

Goal: Move from “good for a small passionate group” → “good for normal people,” and make sure your **metrics + costs** are sustainable.

4.1. Open up access

1. Put TestFlight link:
 - On your website.
 - In social posts.
 - In targeted communities.
2. Start tracking:
 - Installation rate.
 - Onboarding completion.
 - 7-day retention (do they come back?).

4.2. Polish + scale

1. Performance

- Make sure:
 - Cold start is reasonable.
 - Scrolling and swiping stays smooth, even with many emails.
- Fix any p95/p99 latency issues on summary + actions.

2. Confusion killers

- Review your analytics funnels:
 - Where do people drop off?
 - Where do they open modals and then cancel?
- Use that to:
 - Rename actions.
 - Add small tooltips or labels.
 - Simplify steps.

3. Guard sensitive flows

- Anything involving money (pay invoice, subscriptions):
 - Extra careful testing.
 - Clear copy & confirmation screens.
- Ensure:
 - No action fires without explicit user tap.
 - No AI auto-executes anything.

4.3. Prepare for monetization (even if you don't turn it on yet)

- Implement StoreKit 2 flows and **hide them behind a feature flag**.
- Decide:
 - Which features are free vs premium.
 - What the “hook” is for free users (e.g. summaries + a couple actions).
- Make sure:
 - The app still feels valuable in free mode.

Exit Criteria for Phase 4

- 500–2,000 active beta users.
 - 7-day retention trending upward (even if small).
 - Core promise feels true in user language (people say “This saves me time / mental load”).
 - Crash rate is low and stable.
 - You are comfortable turning on payments soon.
-

Phase 5 – Launch Preparation & Launch (2,000+ users)

Goal: Turn a solid public beta into your **1.0 launch**.

5.1. Launch hardening

1. **Bug bash**
 - Run a focused “bug bash” with internal + power users.
 - Triage everything:
 - P0: must fix pre-launch.

- P1: top priority in first post-launch sprints.
- P2+: backlog.

2. Accessibility & edge cases

- VoiceOver basic flows.
- Dynamic type / large font.
- Offline / flaky network behavior.

3. Metrics readiness

- Dashboard for:
 - Daily active users.
 - Summaries requested + helpfulness.
 - Actions completed.
 - Crashes / error rates.
 - AI cost per user.

5.2. Launch playbook

1. Choose a launch date and lock it.
2. Update:
 - App Store metadata (screenshots, copy, subtitle).
 - Privacy policy & TOS as needed.
3. Plan launch comms:
 - Email to existing beta users ("We're live, here's what's new.")
 - Social + any PR you want.

4. Decide your **launch offer** (if relevant):

- Intro pricing.
- Extra free trial period.
- “Founding beta” perks.

Launch Gate Criteria

You green-light full launch when:

- 48 hours in a row with:
 - No critical crashes.
 - No severe action bugs.
 - AI cost per user in expected range.
 - Summaries & Tier 1 actions are **consistently trustworthy**.
 - You’re comfortable exposing this to people who *don’t know you personally*.
-

What to Prioritize

Right Now

(From where you are)

Given: “*App works but is not good.*”

Here’s the immediate sequence:

1. **Instrument + stabilize (Phase 0)**

- Get one clean TestFlight build with analytics + crash reporting.

2. Hammer reading + summary quality for a tiny group (Phase 1)

- Don't touch all 10 actions yet.

3. Pick 3–4 actions and make them really solid (Phase 2)

- Calendar, Reminder, Track Package, Save Email, for example.

4. Then expand user count + broaden the action set (Phase 3+)

- Don't prematurely push to 500+ users with a messy core.

If you'd like, I can next:

- Turn this into a **checklist-style doc** you can drop into Notion, or
- Create a **beta feedback form template** and an **email script** for each phase.

templates+Emails

-
1. **Beta feedback form template** (questions + fields)
 2. **Step-wise emails** you send at each phase gate
-

Beta Feedback Form Template (Copy Into Google Doc / Typeform / Notion)

Title: “Zero Beta Feedback — Help us get to 1.0”

Estimated time: 3–5 minutes

Your Info (optional, non-identifying)

- **Test group:** (Internal / Friends & Family / Guided Beta / Public Beta)
 - **Days using the app:** (1–2 / 3–5 / 5–10 / 10+)
 - **Primary use case:** (School / Shopping / Travel / Bills / Other)
-

Email Reader Experience

1. How clear and comfortable is reading email in the app? (1-5)
2. Do emails *render correctly?* (Yes/Somewhat/No)

If “No”, briefly describe issues:

- cut off text
- scrolling broken
- formatting lost

- images too large
 - missing elements
3. Describe the **biggest reader friction point** in one sentence (open text)
-

Summary Quality

4. When summaries are shown, how often are they helpful and trustworthy?

- 0-25%
- 25-50%
- 50-75%
- 75-90%
- 90%+

5. Do you *notice hallucinated or incorrect details* in summaries? (Yes/No)

If “Yes”, give an example (open text)

6. On average, how much editing do summaries need before you’d send them?

- None
- Light edits (<10% of text)
- Some edits (~10-30%)
- Heavy edits (30%+)

7. Summary confidence score feels appropriate? (Yes/Too High/Too Low/Unclear)

1-Tap Actions (Top 10)

8. Which action did you try? (multi-select)

- RSVP/Add to Calendar
- Sign Document
- Flight Check-in
- Track Package
- Pay Invoice
- Create Reminder
- Save Contact
- List Extraction
- Compound Actions
- Other

9. Did the action **complete reliably** without crashing? (Yes/No/Partial)

10. Did the modal explain what to do clearly? (1-5)

11. Did anything break or fail silently? (Yes/No)



Stability & Performance

12. Any crashes while swiping, summarizing, or completing actions? (Yes/No)

13. Does the app feel responsive?

- Very fast (<1 sec)
- Fast (1-2 sec)
- A bit slow (3 sec)

-
- Slow (>3 sec)

🎨 Overall Experience & Conviction

14. Would you keep using this app today?

- No
- Maybe if it improved
- Yes, even though it's rough

15. Rate your overall satisfaction (1-5)

16. One feature you'd fix **first** if you were product-owner (open text)

⭐ Final Output

17. **Shout-out for what's good** (1 sentence)

18. **Top 3 things to fix** (bullets, open text)

19. Any final notes? (open text)

1. Internal Dev (Phase 0) — Email Script

Send via: Apple internal build + analytics integration via PostHog.

Email

None

Subject: Join our first internal build of Zero

Hey team – the app *works* but needs conviction and clarity to become genuinely good for parents.

This build is purely to:

- enable TestFlight installs
- collect crash + latency telemetry via PostHog
- expose the rough summary + action registry

Please:

1. Install from TestFlight on any available device
2. Triage at least 20 real emails
3. Try 3–5 actions even if janky
4. Submit feedback via our form

We're not refining the business model yet—just making sure the foundation is solid and observable.

Thanks for being our earliest inner loop.

2. Friends & Family (Phase 1-2) — Email Script

Partner credibility: Gmail handled only through OAuth via Google; signatures via PencilKit.

Email

None

Subject: Be part of our Friends & Family Beta for Zero 

Hey! We're evolving Zero into a calmer inbox for parents, starting with better reading + 2–3 sentence AI summaries + 1-tap actions.

Right now it:

- Works ✓
- But is not good (yet!) ⚠

For the next 3–5 days please:

1. Try reading 10+ emails
2. Use summaries on long threads
3. Try 3 core actions: Calendar, Reminders, Package tracking
4. Tell us when it feels broken or cool
5. Submit feedback via our form

Your edits and overrides *help tune the AI*, but you stay 100% in control—nothing auto-sends.

Thank you for helping us get to 1.0.

3. Guided Parent Beta (Phase 3) — Email Script

Data flow mindset grounded in: existing routing cleanup series, using single ActionRouter patterns.

Email

None

Subject: Zero Beta Wave – Start Here (Just 1 suggestion at a time)

Hi! You're in our Guided Parent Beta cohort. We're currently optimizing for:

- ✓ trustworthy summaries
- ✓ magical 1-tap actions

zero tool switching and crashes

Please:

- Use summary at least 10 times
- Complete 1-tap actions from a single suggestion per card
- Override or edit when you disagree
- Let us know your top 3 friction moments

We'll ship improvements *each beta build* based on your responses—starting with stability, reader comfort, and summary quality.

Thanks for tuning AI with us while keeping human conviction intact.

4. Public Install Beta (Phase 4) — Email Script

Marketing surface: App Store (iOS) listing, primary category locked after this cohort.

Email

None

Subject: Public Beta Access for Zero – Help us battle test 

Hi! You're in our Public Beta for Zero.

This wave focuses on:

- 📌 email reading comfort
- 📌 summary relevance and truthfulness
- 📌 the 10 most useful 1-tap inbox actions
- 📌 performance of swiping and modals
- 📌 cost + latency telemetry

Please:

1. Install from TestFlight or App Store
2. Triage at least 25 real emails
3. Try action buttons 5+ times
4. Share when something breaks or feels oddly good
5. Submit feedback via our form

We'll iterate every 1-week sprint (6–12mo incremental evolution path).

Can't wait to ship the next build with fixes influenced by *your real interactions*.

5. Launch Ready (Phase 5) — Email Script

Final quality handoff thinking from startup exits like Superhuman action tiering, Product Sauna style for PM calls.

Email

None

Subject: 🎉 Zero Launch Window – We're About to Ship 🚢

Hey! We're in the final launch staircase.

Before we fully release we're checking:

- ✓ 48 hours with zero critical crashes
- ✓ Tier 1 actions reliable 98%+
- ✓ summaries feel trustworthy and concise
- ✓ parents can describe the value in their own words
- ✓ performance feels responsive under load
- ✓ we're confident turning on pricing tiers soon

You may still notice rough edges in Tier 2-3 actions—those will evolve post-launch in focused sprints, while we keep the promise tight now.

Thank you for helping shape this product from messy to conviction-led. Next stop: 1.0.

If you want, I can also generate the

Typeform JSON schema

next and a pricing elasticity calculator to plug into your business model evolving plan.

phasing plan 2

clear, step-by-step beta phasing plan from “app works but isn’t good” → *launch*, layered for iterative improvement in TestFlight.

Zero iOS Beta Phasing Plan (TestFlight-Led Evolution)

Phase 0 — Baseline Stability + Tooling (1–2 weeks)

Audience: Internal team

Critical goals

- Crash-free installs
- Observability for AI and actions
- A single clean routing baseline

Step-by-step

1. Create a **dedicated TestFlight group** in App Store Connect called “Internal Alpha”.
2. Integrate crash + event telemetry via PostHog:
 - email_loaded, summary_requested, summary_helpful, action_completed, action_failed, crash_occurred, latency_ms, tokens_used
3. Lock routing to your best current single system (discard any modal routing duplication).
4. Ship a **new TestFlight build** from Xcode to internal users only.
5. Run scripted test:
 - 20+ emails triaged/swiped

- 10+ summaries requested
- 3–5 core actions tapped
- 0 blocker crashes allowed

Exit to advance when

- Installable on multiple devices
 - Triage usable without full UX polish
 - Telemetry pipeline working
 - You have a list of the **top 10 friction issues**
-

Phase 1 — Reader + Summary Clarity (2 weeks)

Audience: 0–20 users (still internal)

Critical goals

- Summaries become *useful*, even if imperfect
- UI stops feeling broken
- Feedback starts tuning AI

Step-by-step

1. Enable summarization with a **real model call**: use a reliable LLM such as a Gemini 1.5 variant via Google AI Studio or your existing backend.
2. Update summary prompt to enforce:
 - 2–3 sentences
 - No fabricated details

- <280 characters *preferred*, <500 max allowed internally
3. Add a simple summary feedback UI:
 - “Helpful”  or “Not helpful” 
 - Open text field to correct summary if needed
 4. Fix the 5 worst **reader friction issues** from Phase 0 (layout, scrolling, modal traps).
 5. Run internal benchmark:
 - 250 summary samples manually rated Good/Meh/Bad
 - Target Bad < 20%, Good > 50% at this stage

Exit to advance when

- Summary helpfulness $\geq 70\%$
 - No hallucination patterns
 - Core UI doesn't trap or crash
 - 1-tap summary override works
-

Phase 2 — Friends & Family Beta (2–3 weeks)

Audience: 20–100 people you trust

Distribution: TestFlight (not public yet)

Critical goals

- Core experience feels like a **real product**, not an internal prototype
- 3–4 actions feel magical
- Summaries trend toward clarity over perfection

- Cost model starts sizing with real telemetry

Step-by-step

1. Invite 20–100 testers via TestFlight link.
2. Expose only **3–4 core actions** in UI (others marked “Coming soon”):
 - Calendar/RSVP via EventKit
 - Reminders via Reminders framework
 - Package tracking via shipping org suggestions
 - Save email / bookmark
3. Improve design tokens and layout primitives in this phase with:
 - 16px/20px/24px headings
 - Standard spacing scale (4/8/12/16/24/32)
4. Run parent trial for 3–5 days:
 - 25+ emails triaged
 - 10+ summaries rated
 - Actions completed 5+ times
5. Review telemetry:
 - Avg AI cost per card (initial target < \$0.03/card at small scale)
 - Latency p95 < 3s
 - Tokens/req logged
6. Tune summary prompt again after real edits.

Exit to advance when

- Summary helpfulness \geq 80%
 - At least 2 actions have \geq 90% silent success
 - <40% summary edit rate
 - Latency stable
 - You understand **actual burn per user interaction**
-

Phase 3 — Guided Parent Beta Waves (3–4 weeks)

Audience: 100–500 parents

Distribution: TestFlight

Critical goals

- Core product promise proven at meaningful volume
- 10 actions (in tiers) validated
- Users feel confident, not confused
- AI reliability increases from real feedback

Step-by-step

1. Create a **waitlist microsite** on your domain (React or equivalent, but you control iOS client).
2. Admit TestFlight waves:
 - Wave 1: 50–100 users
 - Wave 2: 100–200 users
 - Wave 3: up to 500

3. Break your 10 actions into tiers:

- **Tier 1 — Must Be Great**
 - Calendar/RSVP, Track package, Create reminder
- **Tier 2 — Good Enough → Improving**
 - Pay invoice, Save contact, File/snooze email
- **Tier 3 — Optional**
 - Compound flows, Admin ML tools

4. Fix Tier 1 reliability:

- Success \geq 95%
- No silent failures
- No crashes on dismiss

5. Tune summary prompt (v1.3) based on Beta edits:

- Bad < 10%
- Meh < 35%
- Good > 55%

6. Check 3 critical funnels in analytics:

- Install → First swipe
- First summary shown → Rated helpful
- Tier 1 action completed → Silent success

7. Finalize subtitle/primary category options, but keep payments **flagged off** until Phase 5.

Exit to advance when

- Tier 1 actions reliable $\geq 95\%$
 - Summary helpful $\geq 85\%$
 - <30% edit rate trending
 - Users can describe value in **their own words**
 - You have conviction, even if not polished
-

Phase 4 — Pre-Launch Public Beta (4–6 weeks)

Audience: 500–2,000→10k (managed expansion)

Distribution: TestFlight initially, App Store link staged

Critical goals

- UX feels good enough for public eyes
- Core promise intact
- No bad builds released prematurely
- Infrastructure + AI burn predictable
- Free tier feels useful
- Payments staged for launch flag

Step-by-step

1. Open TestFlight public group.
2. Expose **all 10 actions**, but visually label by tier if still rough.
3. Fix p95 performance + crash spikes.
4. Improve empty/error states:

- “No emails” state feels helpful
 - “We’re not confident here” state doesn’t fabricate
5. Test money flows carefully under flags:
 - StoreKit 2 UI present but hidden until ready
 6. Begin mock marketing push targeting trial:
 - Test install CTA wording
 - Track CPA, not impressions
 7. Prepare a new repo blueprint + equity hire ramp, but don’t hire 10 modal owners until metrics show which to prioritize.

Exit to advance when

- 7-day retention trending up ($\geq 35\%$ is strong for Phase 4)
- Summary edit < 28%
- Tier 1 reliability $\geq 98\%$ silent success
- Crash <1 per 250 sessions
- Latency solid
- Users describe product value without confusion

Phase 5 — Launch (2,000+ parents validated)

Launch gate only when

None

Given summaries feel concise, truthful, and helpful $\geq 92\%$
And classifier accuracy $\geq 90\%$

And Tier 1 actions reliable 98%+ without crashes or silent failures
And latency <2s local / <3s cloud p95
And 48 hours with zero critical errors
When we expose App Store metadata, pricing tiers, and beta users
Then product is ready for 1.0 launch and follow-up sprints through month 12.

💰 Suggested Budget Telemetry for Foundational Improvements (early sizing)

Area	Startup logic	Est.
Summary+Classifier tuning	1–2 weeks internal	\$15–45k
Infra instrumentation + crash stack	Cloud Run + Build + logging	\$24–66k/6mo
Tier 1 modal polish (3–4 actions)	2 sprints	\$25–90k (payroll effort approximation)
Design token + microsite + comms	1 sprint	\$15–40k
Public Beta expansion + marketing soft push	6 weeks	\$120–250k initial

(You will replace these as real telemetry tightens your model in business model artifacts.)

If you want, I can next generate:

- A **6–12 month iteration calendar**
- Updated **feature flag schedule**
- Action tiers microcopy suggestions
- A Notion epic table you can copy

Would you like that?

combined plan

complete step-by-step beta evolution staircase with TestFlight, plus your requested feedback form and update-email templates.



Beta improvement principles

You only graduate a step when it meets the bar below:

- **True > flashy**
 - **One magical suggestion at a time**
 - **Silent success > noisy options**
 - **Parent override always wins**
 - **Observability from day 1**
-

Phase 0 — Clean baseline, safety, telemetry (Weeks 1–2)

Goal: Make app safe to test, instrumented, and stable.

Steps

1. Fix install blockers and showstopper crashes.
2. Generate a clean build number + version in Xcode.
3. Create TestFlight group in App Store Connect:
 - Group name: **Internal Alpha**
4. Add telemetry with PostHog:

None

```
email_loaded, email_opened, summary_requested,  
summary_helpful, summary_not_helpful, summary_edited,  
action_suggested, action_completed, action_failed,  
latency_ms, tokens_used, crash_occurred
```

4.

5. Ship build to TestFlight and test with **0–20 internal users**.

6. Run **script test** in physical devices:

- 20–30 real emails swiped
- 10–15 summaries rated
- 3 core modal actions tried

Exit criteria to graduate

- Crash-free install
 - Summaries return, even if low-quality
 - 2 actions complete without silent failure
 - Telemetry logs function
 - You have 10 friction issues ranked
-

Phase 1 — Summary + reader comfort overhaul (Weeks 3–5)

Goal: Make summaries and reading not feel broken.

Steps

1. Rewrite summary prompt (v1.1):
 - 2–3 sentences
 - Prefer \leq 280 chars, max 500 internally
 - Never fabricate
 - Say “I’m not sure” explicitly
2. Improve reader primitives:
 - Fix text cut-off and scroll jank
 - Normalize fonts & spacing
 - Fix modal-dismiss traps
3. Add **in-app summary feedback** Yes/No
4. Run internal benchmark batch review:
 - 150–250 summary samples manually rated Good/Meh/Bad

Exit criteria to graduate

- \geq 70% helpful votes
- Hallucination patterns explicitly mitigated
- Reader stable
- Modal UX not trapping

Phase 2 — Friends & Family Beta (Weeks 6–9)

Goal: Small real-world cohort, limited surfaces.

Steps

1. Create TestFlight group:
 - Group name: **Friends & Family Beta**
 - Add 40–120 invited users
2. Only expose 3–4 core actions in UI:
 - RSVP/Add to Calendar
 - Track Package
 - Create Reminder
 - Save Contact
3. Tune design tokens across all views:
 - Heading sizes: 20 / 24 / 32 max scale
 - Spacing scale: 4 / 8 / 12 / 16 / 24 / 32 / 48 max
4. Run 3–5 day trial script
5. Review telemetry:
 - Cost/card trend
 - Latency/ms
 - Action success p95
6. Refine summary prompt again from live edits

Exit criteria to graduate

- $\geq 80\%$ summary helpfulness

- 2 actions silent-success $\geq 90\%$
 - Edit rate $< 40\%$ and trending down
 - Stable latency
-

Phase 3 — Guided Parent Beta (Weeks 10–14)

Goal: 100–500 realistic users, wave improvements.

Steps

1. Create microsite invite landing for Beta optics
2. Create TestFlight group:
 - Name: **Guided Parent Beta**
3. Admit in 3 waves:
 - 75 users
 - 200 users
 - 500 users max
4. Tier the full 10 actions:
 - **Tier 1:** Calendar, Package Tracking, Reminder
 - **Tier 2:** Invoice Pay, Contact Save, File/Snooze
 - **Tier 3:** Compound + ML Admin
5. Polish Tier 1 modals to silent-success bar
6. Improve onboarding + empty/error states

7. Tune summary and classifier weights from friction
8. Run funnel analytics audit

Exit criteria to graduate

- Tier 1 reliability $\geq 95\%+$
 - Summary helpfulness $\geq 85\%+$
 - Edit < 30% trending
 - Crash rate < 1/250 sessions
 - Users can explain value clearly
-

Phase 4 — Public Beta + pre-launch confidence (Weeks 15–24)

Goal: 500–2,000 → 10k installs managed, stable UX.

Steps

1. Create TestFlight group:
 - Name: **Public Beta**
2. Expose 10 curated actions but keep:
 - 1 suggestion per card max
 - Confidence label on modal headers if < 4.0
3. Bug bash performance + cost telemetry
4. Hide or flag purchase flow until reliability validated

5. Begin marketing tests (installation CPA, not opinions)

Exit criteria to graduate

- 7-day retention \geq 35% trending up
 - Summaries \leq 280 reliably for \geq 90%+
 - Tier 1 actions reliability \geq 98%+
 - 48-hour stability freeze: no criticals
-

Phase 5 — 1.0 Launch (Graduation after 48h freeze)

Goal: Launch, then incremental sprints to month 12.

Launch Gate

None

If:

- summary helpfulness \geq 92%
- classification \geq 90%
- Tier 1 actions reliability 98%+
- latency < 2s / 3s cloud p95
- 48h freeze with zero criticals

Then: launch as 1.0

Beta Feedback Form Template (Production-grade)

Fields

- Group: (Internal / F&F / Guided / Public Beta)
- iOS version: (autofilled optional field)
- Device: (model name optional field)

Questions

1. **Reading comfort 1–5**
2. **Rendering issues?** (Yes/No/Some)
3. **Summary helpfulness % bucket**
4. **Hallucinations?** (Yes → give example)
5. **Edit rate?** (None/Light/Some/Heavy)
6. **Modal reliability? 1–5**
7. **Any silent failures?** (Yes/No)
8. **Crashes?** (Yes → when?)
9. **Latency feel** (Fast/Okay/Slow)
10. **Would you keep using?** (Yes/Maybe/No)
11. **One thing to fix first** (open text)
12. **Top 3 issues** (bullets, open)
13. **One sentence praise** (open)



Email scripts for each beta phase

Phase 0 — Internal Alpha

None

Subject: Internal Alpha – Zero needs conviction

Hi team,

The app installs and runs, but the experience and output quality are not good yet.

Our focus this build:

- 1) stability + crash elimination
- 2) summary + classification wiring
- 3) 3 core actions tested
- 4) full telemetry flowing

Please triage 20–30 real emails and submit 1–2 feedback forms.

Thanks for being our first loop.

Phase 1 — Summary & Reader UX pass

None

Subject: Summary + reader overhaul in progress

Hi team,

We've pushed a new summarization prompt and UI primitives for reading comfort.

Now testing:

- summary helpful votes
- no hallucination details unless stated as uncertainty
- modals that trap are now fixed (or being fixed)

Please request 10–15 summaries and rate them in the form.

Help us push Good >50%.

Phase 2 — Friends & Family Beta

None

Subject: Friends & Family Beta invite for Zero 🖤

Hi,

Thanks for agreeing to help us improve Zero.

We're evolving the app step-by-step with real TestFlight users.

This build limits to 3 core actions so we can make them reliable and magical.

Try:

1. read 10+ emails
2. tap summary on long threads
3. complete 5+ actions

Share where it breaks. Override freely.

You stay in control. We tune AI with your edits.

Phase 3 — Guided Parent Beta Waves

None

Subject: Guided Parent Beta – 1 suggestion at a time

Hi, you're in Wave {#}.

We are currently tuning for:

- trustworthy 2–3 sentence summaries ≤280 chars
- at least 3 magical, reliable actions
- silent success and graceful fallback
- telemetry visibility

Please complete 25 emails triaged this week.

Tell us your top 3 friction moments.

Thanks for helping evolve the model.

Phase 4 — Public Beta access

None

Subject: Public Beta – Help us battle-test to 1.0

Hi,

You're in our broader beta cohort now.

We are preparing launch confidence by testing:

- all 10 actions by tier
- 98%+ reliability on core
- summary helpfulness Good >70%
- edit <30% trending
- 48h freeze stability

Please share feedback and invite 2 parents if you'd like.

Thanks for shaping this category with us.

Phase 5 — Launch Prep

None

Subject: Launch Ready Check – 48 Hour Stability Freeze

Hi team, we're in our 48-hour freeze to reach 1.0.

If you see crashes or silent failures, mark them P0 in the form and flag immediately.

We'll follow-up next sprint for Tier 2 improvements.

Thank you for helping layer conviction into the product.

Next stop: App Store 1.0 

If you'd like, I can also create this as a **formatted DOCX file** for you.

Would you like me to generate it now?