



Indevaro

App Growth Decision Playbook

Choose your next move from evidence, test it fairly, and learn from the result

App Growth Decision Playbook

Choose your next move from evidence, test it fairly, and learn from the result

Version 1, editorial revision v1-r4, reviewed 23 October 2026. Sources checked 23 October 2026. Published by Indevaro; written by Indevaro editorial.

Every example in this book is fictional. App names, numbers and outcomes are invented for teaching; any resemblance to a real app is unintended.

Text, worksheets and examples © 2026 Indevaro. All examples are fictional and written by Indevaro; no third-party text, images or customer data are included. Platform names (App Store, App Store Connect, Google Play, Play Console, Reddit, X, Facebook) belong to their owners and are used only to refer to their products; no endorsement is implied. Readers may print and copy the worksheets for their own use and share unmodified copies of the PDF free of charge with attribution. Any other reproduction requires written permission.

Read it on the web, with the same text: indevaro.com/playbook/app-growth. Fonts: Instrument Sans, Newsreader and STIX Two, under the SIL Open Font License 1.1.

Contents

1	How to choose your next app growth experiment	4
2	Why is my app not getting downloads?	9
3	App store keyword research and discoverability	14
4	How to improve app store listing conversion	19
5	Downloads but no paying users: what to check	25
6	Did an app update hurt growth? Check crashes and ANRs	29
7	App competitor analysis: listings, reviews and positioning	34
8	How to measure an app growth experiment	39
9	How to market your app after launch	44
10	How to prioritize growth work across your apps	49
11	How to learn from app growth experiments	53
12	Your next two hours of app marketing	57

Worksheets

1	Decision card	62
2	Missing-data map	64
3	Discoverability source checklist	66
4	Test brief	68
5	Monetization diagnostic tree	70
6	Release regression triage	72
7	Review-theme sheet	74
8	Measurement card	76
9	Channel experiment card	78
10	Two-hours portfolio sheet	80
11	Decision journal	82
12	Your next two hours	84
	Source ledger	89

1.

How to choose your next app growth experiment

In this chapter: you will learn to split every growth question into three parts — what you observed, what you suspect, and what you decide — and to write them on one card before you touch your listing, your price or your code. The card also tells you when the best decision is to change nothing yet.

Every example in this book is fictional. App names, numbers and outcomes are invented for teaching; any resemblance to a real app is unintended.

The problem

It is Monday morning. You open App Store Connect or Play Console and a number is lower than last week. Downloads are down. Or trials are flat while you expected them to rise. Within ten minutes you have a theory ("the new screenshots are worse", "a competitor launched", "the algorithm changed") and within an hour you are redesigning something.

Most indie developers do not lack ideas. They lack a pause between *seeing a number* and *changing the product*. Without that pause, three things go wrong:

1. You react to noise. Small apps have small numbers, and small numbers move a lot on their own.
2. You fix the wrong thing. A drop in downloads can start in discovery, on the product page, in a release, or in the calendar. Each needs a different response.
3. You cannot learn from what you did. If you changed three things in the same week, you will never know which one mattered.

What is going on

A useful growth decision has three layers. Keep them apart, in writing.

Observation (fact). Something you can point to, with its source, its time range and its definition. "Total downloads in the US App Store were 412 last week and 503 the week before, per App Store Connect." An observation has no *because* in it.

Hypothesis (a possible explanation). A statement that could be true or false and that some data could check. "Fewer people saw the app in search that week." "The new first screenshot converts worse." Always write at least two competing hypotheses, including a boring one ("normal week-to-week variation", "a holiday in the main market"). If you can only think of one explanation, you have not looked yet.

Decision (what you will do next). One of four kinds:

- **Act** — make a specific change, because the evidence is strong enough and the cost of being wrong is low.
- **Test** — make a change in a way that lets you compare (a store-listing experiment, a custom product page, a staged rollout).
- **Check** — gather the one missing piece of evidence that would decide between hypotheses.
- **Wait** — change nothing, set a date to look again, and write down what would make you act.

"Wait" and "check" are real decisions, not failures. They are often the best use of your time, because they protect you from undoing something that was working.

Where the numbers come from matters

Before you trust an observation, know how the platform defines it. Apple, for example, counts an *impression* when your app is seen for more than one second in the Today, Games, Apps or Search tabs, and its *conversion rate* is total downloads and pre-orders divided by unique-device impressions — which includes redownloads [S05]. Some App Store Connect usage figures only include users who agreed to share data with developers, and metrics are hidden until a minimum volume is reached [S05]. A number you misunderstand is not evidence; it is a guess wearing a label.

Three words to keep in your head

- **Correlated** means two things moved together. Downloads fell the same week you changed screenshots.
- **Caused** means one thing made the other happen. You can only say this when you have ruled out the other explanations or run a fair comparison.
- **Noise** is movement you would have seen even if nothing had changed. On a small app, noise is often larger than the effect of any single change.

This book returns to these words often. Chapter 8 shows how to measure an intervention without fooling yourself.

How strong is strong enough?

There is no universal threshold. Ask two questions:

1. **How many independent kinds of evidence point the same way?** A dip in downloads *and* a drop in impressions *and* a lost keyword position are three views of one story. Two dashboards showing the same underlying event are one kind of evidence, not two.
2. **What does it cost to be wrong?** Updating promotional text on the App Store can be done at any time without a new app version [S01]; being wrong costs little. Changing your price or your app name is harder to reverse and deserves more evidence.

Worked example

Fictional example · E1 · Hushlog · tags: ios, no-intervention

Situation. Hushlog is a small iOS sleep journal by a solo developer. Its goal this quarter is more downloads. On Monday the developer sees that total downloads in the United States were 412 last week, against 503 the week before — an 18% drop. The first screenshot was replaced two weeks ago, at the start of the week that had 503 downloads. The instinct is to revert it.

Observations (all from App Store Connect, US, same definitions):

- Total downloads: 503 → 412 week over week.
- Impressions (unique devices): down by a similar proportion, about 17%.
- Conversion rate (downloads ÷ unique-device impressions): roughly flat, within a few tenths of a point.
- The week before the screenshot change had 489 downloads; the first week with the new screenshot had 503.
- Last week included a long holiday weekend in the US.

Hypotheses:

1. The new screenshot converts worse. *Weak*: conversion did not drop; the loss is in how many people saw the app, which the screenshot does not affect much for people who never reach the page.
2. Fewer people searched for sleep-related apps last week (holiday, seasonality). *Plausible, not proven*.
3. Hushlog lost visibility in search for its main terms. *Possible; needs a check of search-sourced impressions*.
4. Normal variation. With about 400–500 downloads a week, a swing of this size is not unusual for this app (the developer's own last 12 weeks ranged from 398 to 540).

The trap. The screenshot change is recent and memorable, so it gets the blame. But the evidence points *before* the product page: fewer people saw the app, and those who saw it downloaded at the same rate.

Check performed (20 minutes). The developer opened the App Store Connect acquisition view, filtered to App Store search as the source type [S23], and compared the same two weeks. Search-sourced impressions fell by about the same share as total impressions; browse-sourced impressions fell too. No single source collapsed.

Decision: *Wait*. Keep the new screenshot. No listing change this week. Re-check in two weeks against the same weeks and against the 12-week range. Act only if (a) conversion rate falls below the 12-week low for two consecutive weeks, or (b) search impressions keep falling while browse impressions recover — which would point to a visibility problem worth a keyword check (chapter 3).

What the decision card looked like (abbreviated):

Field	Entry
Goal	More downloads (US)
Observation	Downloads -18% w/w; impressions -17%; conversion flat
Hypotheses	Holiday/seasonality; search visibility; screenshot (weak); noise

Field	Entry
Evidence strength	One kind (App Store Connect acquisition); no conflict
Decision type	Wait
Re-check date	In two weeks (Monday)
Would act if	Conversion below 12-week low for 2 weeks, or search-only impression decline

Do this now

1. **Pick one number that worried you this week.** Write it as an observation: metric name, value, comparison value, time range, market, source.
2. **Look up its definition** in the console's help for that metric. Note anything surprising (redownloads included? opt-in only? minimum thresholds?).
3. **Write three hypotheses**, including one boring one (calendar, noise, a definition change).
4. **For each hypothesis, name one piece of data that would weaken it.** If you already have that data, look now.
5. **Choose a decision type:** act, test, check or wait. If you choose act or test, write the smallest version of the change.
6. **Set a re-check date and a trigger.** "I will act if..." in one sentence.
7. **Fill the decision card** (worksheet below) and keep it where you will see it on the re-check date.

Don't change anything yet when...

- You cannot say which step of the funnel moved (chapter 2). Fix the diagnosis first.
- The change is within your app's own normal range over the last 8–12 comparable periods.
- The comparison window includes a holiday, a release, a price change, a featuring or an ad campaign that you have not accounted for.
- The metric's definition or reporting changed recently (chapter 2 describes a real metric change on Google Play).
- You changed something else in the same period and cannot separate the effects.
- The only evidence is one review, one tweet or one competitor's move.

Worksheet

Decision card — one page per decision: observation, hypotheses, evidence, decision type, re-check date and trigger.

Optional: if you want a second opinion based on your public listing, the free Next Move tool reads your public store listing (on the stores it supports) for one market and goal and returns one evidence-backed move — or a check, when the public evidence is not enough. The card above works without it.

Sources

- S01 — Creating your product page (Apple Developer) — <https://developer.apple.com/app-store/product-page/>
- S05 — Metric definitions, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/reference/metrics-definitions/>
- S23 — Acquisition, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/acquisition/acquisition/>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=positioning can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

2.

Why is my app not getting downloads?

In this chapter: a plain map of the path from "someone sees your app" to "you get paid", using the platforms' own definitions. You will find the step that actually changed, and list what you cannot see — before anyone, including you, starts guessing.

The problem

If your app is not getting downloads, first separate **too few people reaching it** from **people reaching it but not choosing it**. If downloads are steady but revenue is flat, follow the next steps of the funnel instead. Rewriting a listing cannot fix every step.

A quick diagnosis before you change anything

What you can observe	First useful check	What it does not prove
Very little store traffic	Confirm availability in the chosen country; inspect discovery by source and the queries you intended to serve.	That the screenshots are bad.
Traffic holds while downloads fall	Check the same market, source and metric definition; then inspect the listing promise and recent releases.	That one asset caused the change.
Downloads hold while new paying users fall	Follow activation, trial and payment steps with your own analytics.	That you need more acquisition spend.
The app is too new for a stable comparison	Record a baseline, test the installation path and gather structured reactions from the intended audience.	That a few quiet days establish demand or lack of demand.

Pick one market and comparable dates. Keep observations separate from explanations. A listing review can help you inspect the public message, but it cannot supply your private traffic or revenue figures.

"Growth is flat" is not a diagnosis. Flat revenue can come from fewer people finding the app, fewer visitors downloading, fewer new users getting to the point where the app is useful, fewer trial users paying, more subscribers cancelling, or a payment failure. Each of those has a different fix, and most fixes for one step do nothing for the others.

Indie developers often look at the step they are most comfortable with. Designers look at screenshots. Engineers look at crashes. Marketers look at keywords. The result is a lot of work on a step that was never broken.

What is going on

The funnel, one step at a time

Think of five steps. Each step has a count and, between steps, a rate.

1. **Seen.** People see your app somewhere in the store: search results, browse lists, a product page.
 - *App Store:* an **impression** is counted when your app is viewed for more than one second in the Today, Games, Apps or Search tabs; product page views are included [S05].
 - *Google Play:* Play Console reports **store listing visitors** — unique users who visited your listing without already having the app installed on any device [S12]. The default report includes new and returning users, and a user is counted once per day per dimension, so this is not simply "product page views by new people" and does not map directly to Apple's impressions.
2. **Downloaded.** People install.
 - *App Store:* **first-time downloads** and **redownloads** are separate; **total downloads** is their sum [S05]. Apple's **conversion rate** is total downloads and pre-orders divided by unique-device impressions [S05] — so it includes redownloads and it starts from "seen", not from "visited the page".
 - *Google Play:* the page says that from June 2026 primary metrics are based on unique clicks and that from July 2026 store listing reports focus on **intent clicks** on Install, Open or Pre-register [S12]. The **click-through rate** (clicking visitors ÷ store listing visitors) replaces the older "conversion rate". A click is intent, not a finished install. The older acquisition figures are still available in Play Console's Statistics page and CSV exports [S12].
3. **Activated.** New users reach the moment the app is useful to them: the first journal entry, the first budget, the first completed workout. No store console measures this for you. It lives in your own analytics, backend or database — if you have it.
4. **Paying.** Users start paying: a purchase, a subscription, or a trial that converts. On the App Store, **paying users** are unique Apple Accounts that paid for the app or an in-app purchase, and **proceeds** are the estimated amount you receive after taxes and Apple's commission [S05]. Subscription apps also have introductory offers such as free trials; a customer can redeem one introductory offer per subscription group [S09].
5. **Staying.** Users keep paying or keep coming back: renewals, retention, repeat purchases. Store consoles offer some of this (for example Apple's subscription cohort views [S09]); much of it requires your own data or a subscription analytics tool.

Read the rates, not only the counts

A count can fall because the step before it fell. If fewer people see the app, fewer people download it even if your product page is as persuasive as ever. So for each step, ask two questions:

- Did the **count** change?
- Did the **rate from the previous step** change?

If the count fell and the rate held, the problem is upstream. If the rate fell, look at that step.

The step you cannot see is still a step

Every developer has blind spots. Common ones:

- You have no analytics inside the app, so step 3 is invisible.
- App Store Connect usage figures only include users who opted in to share data with developers [S05]. They describe that group, not everyone.
- Your subscription tool shows revenue for a whole project that contains several apps.
- You changed the price and the paywall in the same release.

A blind spot is not a reason to guess. It is a line on your **missing-data map**: which step, what you would need to see it, how long it would take to get, and whether the decision can wait.

Definitions change, and charts do not warn you

Consoles rename and redefine metrics. Google Play's June/July 2026 change from "conversion rate" to click-through rate [S12] is a recent example: a chart that spans the change can show a step that no user caused. Before you compare two periods, confirm that the metric meant the same thing in both — and where the platform keeps the old metric (Play still offers store listing acquisitions in Statistics and exports [S12]), use it to keep one continuous series.

Worked example

Fictional example · E2 · Brightmoss · tags: android, measurement-change, no-intervention

Situation. Brightmoss is an Android flashcards app for language learners, built by two developers. Goal: more paying users. In early August they look at Play Console and see that the headline rate on their store listing report is noticeably higher than in May. One of them wants to credit the new feature graphic uploaded on 1 July; the other wants to put the saved effort into a paywall redesign.

Step-by-step funnel check (Play Console, all countries, May vs August):

Step	May	August	Note
Store listing visitors	9,800	9,650	About flat
Store listing acquisitions (Statistics page, same definition in both months)	2,300	2,270	About flat
Headline rate on listing report	"conversion rate"	"click-through rate"	Metric definition changed in June/July 2026 [S12]
New installs in the developers' own backend (first app open)	2,210	2,190	About flat
Trials started (own backend)	530	540	About flat
Paid subscriptions started (own backend)	88	71	Down about 19%

Observations. Visitors, store listing acquisitions (the continuous series), first opens and trials were stable. The "improvement" on the listing report coincides with Play replacing an outcome metric (acquisitions) with an intent metric (clicks) [S12]. Paid starts fell.

Hypotheses.

1. The feature graphic improved the listing. *Not supported*: store listing acquisitions and first opens did not rise.
2. The listing rate rose because the definition changed. *Supported by the platform's own note*.
3. Something between trial and payment got worse. *Plausible*; it is the only step where the rate moved.
4. Noise. With 71–88 paid starts per month, a swing of 17 is not large. As a rough guide — assuming independent sign-ups and a stable underlying rate — two monthly counts around 80 can differ by about ± 25 (roughly three times the square root of 80) from chance alone (chapter 8 explains the rule). This is a sanity check, not a significance test.

Missing-data map entry. Trial-to-paid by trial start week is not tracked; the team only has monthly totals. They cannot tell whether the drop is concentrated in one week (a bug, a billing issue) or spread evenly (noise, a seasonal pattern).

Decision: *Check, then wait.* Do not credit the feature graphic, do not redesign the paywall. Keep the continuous series from store listing acquisitions in Statistics and exports [S12], and start a new baseline only for the click-through chart. Add a weekly "trials started → paid" cohort to the backend (about one hour of work). Revisit in four weeks; move to chapter 5's diagnostic tree only if the weekly cohort rate stays below May's monthly trial-to-paid rate ($88 \div \sim 530$ trials, about 17%) for three consecutive weeks.

Do this now

1. **Draw your five steps** for one app, one market: seen, downloaded, activated, paying, staying.
2. **For each step, write where the number comes from** (console, your backend, a subscription tool) and **its exact definition**. Copy the definition from the platform's help page, not from memory.
3. **Fill the last two comparable periods** for counts and rates. Use the same length and, if you can, the same weekdays.
4. **Mark the first step whose rate changed.** That is where you look next. If none changed, the problem is upstream of your data (seasonality, the market) or it is noise.
5. **List your blind spots** on the missing-data map: step, what's missing, how to get it, effort, and whether the current decision can wait for it.
6. **Check for definition changes** in the period: console notes, SDK updates, a new analytics event name.

Don't change anything yet when...

- The first changed rate is in a step you cannot see — fill that gap first.
- A console renamed or redefined the metric inside your comparison window.
- Counts are small enough that your normal range covers the change.
- Several steps moved together in a way one event could explain (a release, an outage, a holiday).
- You would need to change more than one step at once to act on your theory.

Worksheet

Missing-data map — the five steps, their sources and definitions, the rates for two periods, and a list of blind spots with effort and urgency.

Optional: Next Move works from public listing data only, so it cannot see steps 3–5 either; when a move depends on them it says so and suggests a check instead. That is the same rule this chapter asks you to follow.

Sources

- S05 — Metric definitions, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/reference/metrics-definitions/>
- S09 — Auto-renewable subscriptions (Apple Developer) — <https://developer.apple.com/app-store/subscriptions/>
- S12 — Acquisition reports, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9859173>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=downloads can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

3.

App store keyword research and discoverability

In this chapter: how to check whether people can find your app, what each store console can and cannot tell you about search, and why a keyword rank is not the same thing as demand for that keyword. You will finish with a source checklist for your own app.

The problem

Keyword tools make discoverability feel precise. You see that your app is ranked 7th for "budget planner" and 41st for "expense tracker", with a difficulty score and a popularity score next to each. It is tempting to rewrite your subtitle every week to push those numbers.

Three problems hide behind the precision:

1. **A rank is a position, not traffic.** Ranking 3rd for a phrase almost nobody types brings almost nobody.
2. **Third-party popularity scores are estimates.** They are useful for comparing terms with each other; they are not counts of searches, and different tools disagree.
3. **Search is only one way people find apps.** Browse lists, referrals from websites, links shared in communities, ads and brand searches (people who already know your name) all bring visitors. If most of your visitors come from somewhere other than generic search, keyword work is not your first lever.

What is going on

What the stores say about search

Apple says App Store search results depend on **text relevance** — matches for your title, subtitle, keywords and primary category — and on **user behavior**, such as downloads and ratings and reviews [S02]. You control the first part directly and the second only indirectly.

On the App Store your text fields are limited: the app name and the subtitle can each be up to 30 characters, and the keyword field holds 100 characters in total [S01]. Apple's guidelines ask for keywords that describe the app accurately and forbid packing metadata with trademarks, other apps' names or irrelevant phrases [S06].

On Google Play there is no separate keyword field. The app name (up to 30 characters), short description (up to 80) and full description (up to 4,000) carry your text, and Google warns that repetitive or irrelevant keywords can get an app suspended [S13]. Play's metadata policy rules out ranking claims such as "#1" or "App

of the year" in the title, icon and developer name, and lists them among common violations in other listing text and images [S14].

What the consoles can show you

This is the most useful part of the chapter, because the consoles already hold better evidence than most keyword tools.

App Store Connect. The acquisition view splits impressions, product page views and downloads by **source type**: App Store search, App Store browse, app referrer, web referrer and others, filterable by territory and device [S23]. Note that App Store search includes views and downloads from ads shown in search results [S23] — if you run Apple Ads, part of your "search" traffic is paid.

Play Console. The store listing report can be filtered by traffic source, country, language and **search term** — the term a user searched before visiting your listing [S12]. Watch the definitions closely:

- *Google Play search* covers visitors who searched for your app's **name or a closely associated brand** [S12]. It measures people who already know you.
- *Google Play explore* covers people who found you while browsing — home pages, suggestions, charts — **or by searching for a category of apps**, such as "racing game" [S12]. Generic keyword discovery on Play shows up here, not under "search".

That single definition changes how you read a Play chart. A rise in "search" may mean more people heard your name elsewhere; a fall in "explore" may be a visibility problem for category terms.

Rank versus demand

Keep three different questions apart:

Question	What answers it	What does <i>not</i> answer it
Do people look for this kind of app with these words? (demand)	Play's search-term dimension for your own traffic [S12]; search-term data from your own search ads, if you run any; third-party estimates <i>as a rough comparison only</i>	Your rank for the term
Does the store show my app for these words? (visibility)	Your rank, checked in the market and language that matter	Popularity scores
Do people who see me for these words choose me? (conversion)	Conversion by source in App Store Connect [S23]; clicks by search term on Play [S12]	Rank or popularity

A good discoverability decision needs at least two of the three. A high rank for a low-demand term is a vanity number. High demand where you rank 60th is an opportunity only if your app really matches what those people want.

Custom pages widen your options

Both stores let you show a different page to different visitors. Apple's custom product pages (up to 70) can have their own screenshots, promotional text and app previews, their own keywords, and unique URLs [S03]. Google Play's custom store listings (up to 50) can target countries, search keywords, a unique URL or a Google Ads campaign [S11]. For discoverability this means you do not have to bend your main listing towards one audience to test a second one.

Worked example

Fictional example · E3 · Fernpath · tags: android, discoverability

Situation. Fernpath is an Android app for planning hikes with offline maps, made by a solo developer. It is available in several English-speaking countries, with most installs from the UK. A keyword tool shows Fernpath dropping from 5th to 14th for "offline hiking map" in the UK over three weeks. The developer plans to rewrite the app name to "Fernpath: Offline Hiking Map" and the short description around the same phrase.

Observations (Play Console, UK, last 28 days vs the 28 days before):

- Store listing visitors: roughly flat (−3%).
- By traffic source: *Google Play search* (brand) slightly up; *Google Play explore* down about 9%.
- Search-term dimension: "offline hiking map" brought 4% of visitors in both periods. The largest non-brand terms were "walking routes uk" and "trail planner", which together brought about three times as many visitors, and both held steady.
- Click-through rate on the listing: unchanged within its normal weekly range.

Hypotheses.

1. The rank drop for "offline hiking map" costs meaningful traffic. *Weak*: the term brings a small, unchanged share of visitors.
2. Explore traffic fell because of seasonality (autumn in the UK). *Plausible*; the developer's data from last year shows a similar dip from September.
3. A competitor's update pushed Fernpath down for several category terms at once. *Possible*; explore fell, but the two biggest category terms held.

What would change the decision. If "walking routes uk" or "trail planner" lost rank *and* visitors, that would be a real visibility problem for terms that matter to this app.

Decision: *No rename. Test one thing.* Keep the app name, which is also what brand searchers type. Create one Play custom store listing targeted at the search keyword "offline hiking map" [S11], whose first screenshot shows offline map download — a feature the app has. Its numbers show how people searching that term respond to that page; they cannot be compared fairly with the default listing, because the audiences differ. To judge the screenshot itself, run a store listing experiment on that custom listing — Play allows experiments on custom listings as well as the default one [S10] — with the current screenshot as control. Check explore traffic against last year's seasonal pattern at the same time. The rename is off the table unless explore keeps falling after the seasonal window *and* the main category terms lose visitors.

Do this now

1. **Split your visitors by source** for one market and the last 28 days: App Store Connect source types [S23], or Play traffic sources [S12]. Write down the share of each.
2. **If generic search/explore is under about a third of your visitors**, note that keyword work is probably not your first lever; check chapter 4 (conversion) and chapter 9 (distribution) first. This cut-off is a prioritizing habit, not a rule from either store.

3. **List your top 10 non-brand terms** from Play's search-term dimension, or your best estimate on iOS (search terms from your own search ads if you run any; otherwise a keyword tool, labelled "estimate").
4. **For each term, fill three columns:** demand (source), your rank (market, date), conversion if known.
5. **Check your metadata for rule problems:** competitor names or irrelevant terms in keywords [S06], ranking claims on Play [S14], repeated keywords [S13].
6. **Pick at most one change** — a keyword swap, a subtitle test on a custom page, or a Play custom listing — and write its test brief (chapter 4).

Don't change anything yet when...

- The term you lost rank for brings a small share of your visitors, or you do not know its share.
- The only evidence is a third-party popularity score.
- Brand search is your largest source; your name is working — renaming risks it.
- The drop coincides with a known seasonal pattern or a store-wide change you cannot see into.
- You changed metadata in the last few weeks and the stores have not settled yet; measure the last change before making the next.
- The only way to rank for the term would be to describe a feature the app does not have.

Worksheet

Discoverability source checklist — sources, their definitions, what each can and cannot tell you, and a demand/visibility/conversion grid for your top terms.

Optional: Next Move can compare your public listing with one competitor's for a chosen market and goal. It reads public listing data only (on the stores it supports) — text, visuals, ratings and a sample of reviews; it does not see your search terms or traffic sources, so it will not claim a keyword is worth ranking for.

Sources

- S01 — Creating your product page (Apple Developer) — <https://developer.apple.com/app-store/product-page/>
- S02 — App Store search (Apple Developer) — <https://developer.apple.com/app-store/search/>
- S03 — Custom product pages (Apple Developer) — <https://developer.apple.com/app-store/custom-product-pages/>
- S06 — App Review Guidelines (Apple Developer) — <https://developer.apple.com/app-store/review/guidelines/>
- S10 — Store listing experiments, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/12053285>
- S11 — Custom store listings, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9867158>
- S12 — Acquisition reports, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9859173>

- S13 — Create and set up your app, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9859152>
- S14 — Metadata policy, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9898842>
- S23 — Acquisition, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/acquisition/acquisition/>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=downloads can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

4.

How to improve app store listing conversion

In this chapter: how to turn "the screenshots feel weak" into a message hypothesis, a comparable test on the App Store or Google Play, and a one-page test brief with a stop rule. You will also learn to estimate, before you start, whether your traffic is big enough for a test to say anything.

The problem

To improve app store listing conversion, start with one visitor question your page does not answer clearly. Show a truthful answer, choose a comparison that reaches a similar audience, and check whether you have enough traffic to learn from the result. More screenshots or a higher audit score are not outcomes in themselves.

A practical listing review checklist

1. **Audience:** can a new visitor tell who this app is for?
2. **Promise:** does the first visible message name a concrete task or benefit, rather than only the feature category?
3. **Demonstration:** do screenshots show the current app doing the promised task?
4. **Consistency:** do the title, first screenshots and description support the same reason to install?
5. **Objections:** is there a repeated question in recent reviews that accurate listing copy could answer?
6. **Measurement:** have you recorded the current assets, audience, metric and date before changing them?

On iOS, product page optimization tests creative assets; a subtitle rewrite is a separate metadata change, not a PPO treatment [S04]. On Google Play, description tests need a localized experiment [S10]. Choose the tool for the change you are actually making.

Your product page is the one growth surface you fully control, so it gets changed a lot. New screenshots, a new subtitle, a new icon, a new first line of the description — often all at once, often without a record of what was there before. A month later downloads are a bit higher or a bit lower and nobody can say why.

The fix is not "never change the listing". It is to change it as a **test of a specific idea about the visitor**, in a way that can be compared, and to know in advance whether the result could be meaningful.

What is going on

Start with a message hypothesis, not an asset

A screenshot is a carrier for a message. Before you touch the design, write the message you believe will work better, and why:

"Visitors from search for 'budget planner' want to see *how quickly they can set up a budget*. Our first screenshot shows a chart of past spending, which answers a different question. If the first screenshot shows setup in under a minute, more of them will download."

A good message hypothesis has four parts: **who** (which visitors, from which source), **what they want to know**, **what the page tells them now**, and **what you will show instead**. It must be true of your app: the promise has to match a feature that exists. Apple's guidelines expect screenshots to show the app in use, not only title art or a login screen [S06]. Google Play's metadata policy lists ranking claims ("#1", "App of the year") and price or promotion text among common violations, and does not allow unattributed testimonials in the description [S14].

The comparison tools you already have

App Store — product page optimization. You can test alternate icons, screenshots and app previews against your original page. A test can have up to three treatments, you choose what share of traffic goes to the test, and a test runs for up to 90 days [S04]. New metadata in treatments must be approved by App Review, and alternate icons must be included in the app binary [S04]. Apple's results show impressions, conversion rate and a confidence level, and Apple suggests waiting until at least one treatment is declared better or worse than the baseline with at least 90% confidence [S04].

App Store — custom product pages. Up to 70 extra versions of your page, each with its own screenshots, promotional text and app previews, and its own URL [S03]. They are not a split test of the same audience; they are a way to show a *different* page to a *different* audience (for example, visitors from a specific community or ad). Apple reports impressions, downloads and conversion for each page [S03].

App Store — promotional text. Up to 170 characters above the description, changeable at any time without a new app version [S01]. Cheap to change, but also low in visibility; do not expect it to explain a large change.

Google Play — store listing experiments. Play distinguishes one default graphics experiment from up to five localized experiments. Default graphics experiments test the icon, feature graphic and screenshots; localized experiments can also test descriptions in up to five selected languages. Each experiment supports up to two variants against the current listing. You choose the audience percentage, which is split equally across variants. You also set a **minimum detectable effect** (the smallest difference you care about) and a **confidence level**; a difference smaller than the minimum detectable effect counts as a draw, and a test can end with "More data needed" [S10]. Experiments stop automatically after six months, and Google recommends testing one asset at a time [S10].

Google Play — custom store listings. Up to 50 listings targeted by country, search keywords, a unique URL or a Google Ads campaign [S11] — the Play counterpart to custom product pages.

Will your test be able to say anything?

This is the question most small apps skip. A test compares two rates. If the rates are low and the visitor counts are small, random variation is larger than any realistic improvement, and the test ends with "no result" after weeks of waiting.

A rough planning rule, with its assumptions stated:

- **Assumptions:** visitors are independent; traffic mix and the underlying rate are stable during the test; you want about an 80% chance of detecting a real difference of the size you name, with a conventional 5% false-alarm rate, comparing one treatment with the original. This is a rough normal approximation for equal, independent arms; use an exact or store estimator for low rates, small counts, unequal allocation, repeated users or devices, or clustered traffic.
- **Rule of thumb:** visitors needed *per arm* $\approx 16 \times p \times (1 - p) \div d^2$, where p is your current rate as a fraction and d is the absolute difference you want to detect.
- **Example:** current rate 4% ($p = 0.04$), you hope to detect a move to 4.8% ($d = 0.008$). $16 \times 0.04 \times 0.96 \div 0.000064 \approx 9,600$ visitors per arm.

This is a planning estimate, not the test's own statistics; use the store's result for the verdict. But if the estimate says you need four months of traffic, you know before you start that the design is unlikely to distinguish a subtle screenshot tweak within the available window. Your options then are: test a **bigger** difference (a different message, not a different shade), send **more** traffic to the test, test on your **largest** market only, or accept that this change will be a judgment call and record it as such (chapter 11).

Apple's product page optimization setup can also estimate the impressions and duration needed to reach a result at 90% confidence [S04] — use it.

Set a minimum run before you look for a winner

Confidence readings wobble while a test is young. If you stop the first day a treatment crosses the line, you will stop on lucky streaks more often than you think (chapter 8 calls this peeking). So write two numbers into the brief: a **minimum run** — at least the planned sample per arm, rounded up to full weeks so every weekday is covered — and a **maximum run** (the tool's limit). Before the minimum run, do not act on the result. After it, apply the store's rule: Apple suggests waiting for at least 90% confidence [S04]; Google Play reports against the confidence level and minimum detectable effect you set [S10]. If the maximum run arrives first, record the estimate and interval or precision information the tool provides; do not infer a small effect from a nonsignificant result, and record the verdict as inconclusive unless the planned precision excludes a meaningful effect.

Keep one change per comparison

If you change the icon, the first screenshot and the subtitle in the same week, any result belongs to the bundle. That can be acceptable — sometimes a coherent new message needs several assets — but then the test brief must say "we are testing message B as a bundle" and nobody may later credit one asset.

Worked example

Fictional example · E4 · Coinmoth · tags: ios, small-sample, listing-test, inconclusive

Situation. Coinmoth is a budgeting app for iOS, built by a two-person team. Goal: more downloads in the US. They set up a product page optimization test with **two** treatments (new first screenshot "Set up a budget in 60 seconds" and a variant with a different color scheme) and allocated **30%** of traffic to the test — so each treatment received 15% of traffic and the original 70% [S04].

Observations after six weeks (App Store Connect, US):

- About 3,000 unique-device impressions per week in total, so roughly 450 per treatment per week — about 2,700 per treatment over the six weeks.
- Conversion rate: original around 4%; both treatments within half a point of it, one up, one down.
- The test result showed no treatment performing better or worse at the suggested confidence level [S04].

Planning check they skipped. At a 4% base rate, detecting a 0.8-point improvement needs roughly 9,600 visitors per arm (rule above). At 450 per week per treatment, that is about 21 weeks — longer than the 90-day maximum [S04].

Hypotheses.

1. The new message works, but the test is too small to show it. *Cannot be ruled out.*
2. The new message makes no meaningful difference. *Cannot be ruled out.*
3. The color variant diluted traffic without testing a clear idea. *Yes — it split the sample for a question nobody had written down.*

Decision: *Stop the test; keep the original; re-run a narrower test.* The result is recorded as **inconclusive**, not as "the new screenshot doesn't work" — and because the test had less than a third of the sample it needed, its interval remains compatible with effects the team would care about. The re-run has **one** treatment (the 60-second setup message, applied to the first two screenshots because the first screenshots can appear in search results when there is no app preview [S01]), **50%** of traffic, and a pre-written stop rule. Before starting, the team checks the rule of thumb again: with ~1,500 impressions per arm per week, about 6.4 weeks gives the needed ~9,600 per arm if the real effect is as large as hoped. So the minimum run is 7 full weeks: no decision before day 49. After that, apply the treatment if Apple reports it better at its suggested confidence level [S04]; if that has not happened by day 90, report the estimate and interval or precision information available and record the test as inconclusive unless the planned precision excludes a meaningful effect.

Do this now

1. **Write one message hypothesis** in the four-part form: who, what they want to know, what the page says now, what you will show instead.
2. **Confirm the promise is true** of the current app version. If it depends on a planned feature, stop here.
3. **Pick the tool:** product page optimization or a Play store listing experiment for "same audience, different page"; a custom product page or custom store listing for "different audience, different page".

4. **Estimate the sample** with the rule of thumb or the store's own estimator. If it exceeds the tool's time limit (90 days on the App Store [S04]; six months on Play [S10]), make the difference bigger or the traffic larger.
5. **Write the test brief** (worksheet below): hypothesis, variant, audience, traffic share, primary metric, minimum and maximum run, stop rule, and what you will do for each outcome — better, worse, inconclusive.
6. **Freeze other listing changes** in the tested market until the test ends.

Don't change anything yet when...

- Your planning estimate says the test cannot reach a result within the tool's time limit, and you cannot make the difference bigger.
- The page is not the step that changed (chapter 2): impressions fell but conversion held.
- A release, a price change or a promotion is planned during the test window.
- The new message depends on a feature users cannot use yet.
- You have not written what you will do if the result is inconclusive.
- Another listing test in the same market is still running.

Worksheet

Test brief — hypothesis, variant, audience, sample estimate, stop rule and pre-committed actions for each outcome.

Optional: one of the materials Next Move can prepare is a screenshot brief or an experiment card based on your public listing and the goal you choose. It describes the message and layout; it does not generate final artwork, and it does not know your traffic, so the sample check above is still yours to do.

Sources

- S01 — Creating your product page (Apple Developer) — <https://developer.apple.com/app-store/product-page/>
- S03 — Custom product pages (Apple Developer) — <https://developer.apple.com/app-store/custom-product-pages/>
- S04 — Product page optimization (Apple Developer) — <https://developer.apple.com/app-store/product-page-optimization/>
- S06 — App Review Guidelines (Apple Developer) — <https://developer.apple.com/app-store/review/guidelines/>
- S10 — Store listing experiments, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/12053285>
- S11 — Custom store listings, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9867158>
- S14 — Metadata policy, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9898842>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=downloads can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

5.

Downloads but no paying users: what to check

In this chapter: how to tell whether "fewer paying users" is an acquisition problem, an activation problem, a paywall problem or a payment problem — using the data you actually have. You will build a diagnostic tree that never asks you to judge a paywall nobody can see.

The problem

Revenue is the number that hurts most when it falls, so it attracts the most dramatic fixes: a new paywall design, a new price, a new trial length, a lifetime option. These are among the hardest changes to evaluate and to reverse. Existing subscribers see prices change, reviews mention it, and every later comparison has a before-and-after split running through it.

Many revenue drops are not paywall problems at all. Fewer people arrived. Fewer new users got far enough into the app to see the paywall. A payment step failed after a release. Or the people arriving were different — returning users, a new country, a new traffic source — and behaved differently.

What is going on

Four places revenue can break

Think of them in this order, because each one limits the next. In practice, check the payment step for errors first — it is quick, and a broken payment step can look like any of the others — then walk from acquisition down:

1. **Acquisition** — how many new people start using the app. Covered in chapters 2–4.
2. **Activation** — how many of them reach the moment the app is useful and, in most apps, the moment they are shown an offer.
3. **Paywall** — of those who see the offer, how many start a purchase or trial.
4. **Payment and renewal** — of those who start, how many complete payment, convert from trial to paid, and keep paying.

The counts multiply. If acquisition fell 20% and everything else held, paying users fall about 20% and the paywall is innocent.

Which data sees which step

Step	Store consoles	Your own data	Public listing
Acquisition	Downloads, first-time vs redownloads [S05]	Installs, first opens	Ratings count, text — not traffic
Activation	Sessions and active devices, for users who opted in to share data [S05]	Onboarding and key-action events	Nothing
Paywall	Nothing about views of your paywall	Paywall-shown and purchase-started events	Nothing
Payment and renewal	Paying users and proceeds [S05]; subscription offers and cohort views [S09]	Purchase results, errors, renewals	Nothing

Two rules follow:

- **If you do not log "paywall shown", you cannot evaluate the paywall.** You can only evaluate the whole path from install to purchase. Add the event before you redesign anything.
- **No one outside your app can evaluate your paywall either.** A public store page does not show it. Any tool, consultant or template that "reviews your paywall" from the listing is guessing. Next Move follows the same rule: from public data alone it can point at acquisition or positioning questions, and for revenue questions it asks for a check that needs your private data.

Composition changes are a common hidden cause

The people who arrive can change without the count changing. Examples:

- More **returning users** (redownloads) after a feature in a newsletter. On the App Store, a customer can redeem one introductory offer per subscription group [S09], so many returning users are not eligible for your free trial and will see a different offer.
- A new **country** where prices are relatively higher or purchasing habits differ.
- A new **traffic source** — a community post or an ad — bringing curious visitors rather than people with the problem your app solves.

Before you blame the paywall, split the paying-user drop by first-time vs returning, by country and by source.

Payment problems look like persuasion problems

If a release broke the purchase flow for some devices, the paywall "converts worse" — but no design change will fix it. Chapter 6 covers how to connect a release, error evidence and a business metric. That is why the diagnostic tree starts with payment errors, before anything else.

Platform rules to keep in mind

On the App Store, unlocking features, subscriptions or premium content inside the app generally has to use in-app purchase [S06]. Apple's current §3.1.1(a) also permits specified external-purchase links; entitlement and storefront conditions apply, and US storefront apps have a broader no-entitlement rule for buttons, external links and calls to action [S06]. Check the current guideline text for your case before you plan a pricing experiment that depends on an external checkout.

Worked example

Fictional example · E5 · Pantrio · tags: ios, monetization

Situation. Pantrio is a meal-planning app with a monthly subscription and a 7-day free trial, available on iOS and Android, built by a solo developer. In August, new paying subscribers on iOS fell to 96 from about 130 in each of the previous three months. Android was stable. The developer is preparing a new paywall with an annual plan up front.

Walking the tree (iOS, August vs May–July average):

Step	Evidence	Result
Acquisition	Total downloads flat (App Store Connect). Redownloads rose from about 15% to about 35% of total; first-time downloads fell.	Composition changed
Activation	Own events: share of new installs creating a first meal plan — flat	No change
Paywall	Own events: paywall shown per activated user — flat; trial starts per paywall view — down	Changed
Payment	Own events: purchase errors — no increase; trial-to-paid rate — flat	No change

Observation. The drop is in trial starts, and it coincides with a jump in returning users. A food blog had featured Pantrio in early August, and many readers were former users.

Hypotheses.

1. The paywall became less persuasive. *Weak*: nothing on it changed, and Android — same design — is stable.
2. Returning users see no free trial, because they already used their introductory offer in this subscription group [S09], and fewer of them subscribe at full price. *Plausible and checkable*.
3. Seasonality. *Possible*; summer months were not lower last year, though.

Check performed (45 minutes). The developer split trial starts by whether the device had installed the app before (own backend flag). First-time users started trials at the same rate as in May–July. Returning users started almost none — consistent with not seeing the trial.

Decision: *No paywall redesign*. Note the August drop as a composition effect from one referral, not a paywall regression. Keep the planned annual-plan idea on the list for a proper test later (chapter 4 rules apply to paywalls too). Separately, consider an offer aimed at returning users — for example, Apple's win-back or promotional offer types [S09] — as its own small test with its own measurement card (chapter 8), after checking eligibility rules in the current documentation.

Do this now

1. **Write your four counts** for the last full period: new users, activated users, users shown the offer, paying users. Mark any you cannot fill.
2. **Add the missing events** before any redesign: "paywall shown", "purchase started", "purchase completed", "purchase failed" with an error reason.
3. **Split the paying-user change** by first-time vs returning, by country and by source.

4. **Check payment errors** around your last release (chapter 6) before you judge persuasion.
5. **Compare platforms** if you have both. A drop on one platform with the same paywall on the other points away from the paywall's design — though platforms differ in audience, prices and offer rules, so it is a hint, not a control group.
6. **Only then** write a paywall or pricing test brief (chapter 4), with a stop rule and a plan for existing subscribers.

Don't change anything yet when...

- You do not log paywall views; you would be changing a step you cannot measure.
- Acquisition or activation changed in the same period.
- The mix of users (returning, country, source) changed.
- A release in the same period touched the purchase flow and you have not checked errors.
- Your monthly count of new paying users is small enough that the change is inside its normal range (chapter 8 has a rough check).
- The change would affect existing subscribers and you have not decided how to treat them.

Worksheet

Monetization diagnostic tree — the four steps in order, the data source for each, and the branch you end up on, including "can't tell yet — add this event".

Optional: if your goal is paying users, Next Move can look at your public listing for positioning issues that affect who arrives. It will not claim anything about your paywall, trial or payment flow, because public data cannot see them.

Sources

- S05 — Metric definitions, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/reference/metrics-definitions/>
- S06 — App Review Guidelines (Apple Developer) — <https://developer.apple.com/app-store/review/guidelines/>
- S09 — Auto-renewable subscriptions (Apple Developer) — <https://developer.apple.com/app-store/subscriptions/>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=paying_users can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

6.

Did an app update hurt growth? Check crashes and ANRs

In this chapter: how to connect three kinds of evidence — a release, error data and a business metric — before you blame marketing for a drop. You will learn how phased and staged rollouts limit the damage, and how to decide whether a fix comes before everything else.

The problem

Growth problems and technical problems look the same on a revenue chart. A crash on launch for one device family, a sign-in failure after an SDK update, a purchase flow that hangs on older OS versions: each shows up as "fewer active users" or "fewer paying users", often days before anyone reads the crash report.

Meanwhile, the usual growth reflexes make it worse. A new screenshot test started during a crash spike will be judged on broken traffic. A price change shipped in the same release will be blamed for — or credited with — whatever the bug did.

What is going on

Three kinds of evidence, one timeline

Put these on the same timeline, with dates and versions:

1. **Release evidence.** Version number, release date, rollout percentage over time, what changed (including SDK and dependency updates, server changes, and remote configuration).
2. **Error evidence.** Crash rate, freeze or "not responding" rate, error logs, failed purchases, failed sign-ins — by app version, OS version and device.
3. **Business evidence.** Active users, sessions, trials, purchases, ratings — by the same version and device splits if you can.

A regression is **plausible** when the error evidence changes at the release date, is concentrated in the new version, and the business metric moves in the same segment. It is **likely** when rolling back or halting the release (or fixing the bug) reverses the business change in that segment. Timing alone is correlation; the segment match is what makes it convincing.

What the platforms give you

Google Play — Android vitals. Play collects stability and performance data from devices whose users allow it. Its core vitals include the user-perceived crash rate and the user-perceived ANR ("application not responding") rate. Play publishes bad-behavior thresholds — 1.09% for user-perceived crashes and 0.47% for user-perceived ANRs overall, and 8% for either on the app's aggregate for a single phone model — and says that above a threshold it may reduce your app's visibility and may show users a warning on your listing [S18]. Treat these as the platform's enforcement lines, not as targets: many regressions hurt users long before they cross them.

Google Play — staged rollouts. You choose the percentage of users who get an update; it does not increase on its own. If you discover a problem, halting the rollout stops any additional users from receiving the update; users who already have it keep it. A new release started before the previous rollout completed goes to the same group of users, depending on its percentage [S19]. If the fix must reach the same affected cohort, consider matching or exceeding the halted percentage after checking the current rollout controls; the source does not make that an absolute minimum.

App Store — phased release. An update can go to users with automatic updates over seven days: 1%, 2%, 5%, 10%, 20%, 50%, then 100%. You can pause for up to 30 days in total, and anyone can still update manually from the store during a phased release [S08].

App Store Connect — crashes. App Store Connect reports crash counts; usage-based figures come from users who opted in to share data with developers [S05]. For detail by device and stack trace, most teams also use a crash-reporting SDK.

Why the rollout controls matter for growth decisions

A staged or phased release is not only a safety net. It is a natural comparison: for a few days, some users run the new version and some the old one, and the platform picks the updated group at random [S19] [S08]. The comparison is cleaner on Google Play; on the App Store the random part covers only users with automatic updates, while anyone who downloads or updates manually gets the new version [S08], so those users are self-selected. If errors and business metrics differ between versions in that window, you have much better evidence than a before-and-after chart. Use the window: look at the numbers by version on day 2 or 3, not only after 100%.

Ranking a fix against growth work

A confirmed regression that affects a meaningful share of users usually comes first, for three reasons: it hurts users now, it corrupts every other measurement you are running, and on Google Play it can affect visibility if it crosses the bad-behavior thresholds [S18]. But "confirmed" matters. An unconfirmed error spike gets an investigation, not a week-long rewrite.

Worked example

Fictional example · E6 · Tallowbird · tags: android, technical-bug

Situation. Tallowbird is a habit tracker for Android, built by a solo developer. Version 4.2 moved reminder scheduling to a new background library. It went out as a staged rollout at 20%. Three days later, the developer notices fewer trial starts on Android and plans to test a new paywall headline.

Timeline and evidence:

Date	Release	Error evidence (Play Console, Android vitals)	Business evidence (own events)
Day 0	4.2 at 20%	—	—
Day 1	4.2 at 20%	User-perceived ANR rate for 4.2 noticeably higher than for 4.1; concentrated in one manufacturer's devices running one Android version	—
Day 3	4.2 at 20%	Same pattern; overall app ANR rate still under Play's 0.47% threshold, but the 4.2 rate on the affected device model is far above 8%. Play assesses the 8% per-phone-model threshold on the app/device-model aggregate over the last 28 days, so this version slice is evidence of risk, not itself a threshold crossing [S18]	Trial starts per active user on 4.2: clearly lower than on 4.1 in the same days
Day 3	—	Six new 1-star reviews mention "freezes when I open reminders"	—

Hypotheses.

1. The new background library causes ANRs on one device family, which stops users before they reach the reminder setup where the trial offer appears. *Strong*: timing, version and device segment match, and reviews describe the symptom.
2. The paywall headline is weak. *Weak*: 4.1 users in the same days, with the same paywall, started trials at the usual rate.
3. Noise. *Unlikely*: the difference is concentrated in exactly the segment with the errors.

Decision: *Halt the rollout and fix first.* The developer halted the 4.2 staged rollout so no additional users receive it [S19], reverted reminder scheduling for the affected OS version in 4.2.1, and released 4.2.1 as a staged rollout at 20% — matching the halted share after checking the current rollout controls. When a new release starts before the previous rollout is complete, Play uses the same group of users as the previous release, depending on the percentage [S19]; matching the share was the team's chosen way to target that affected cohort, not a source-mandated minimum. The check on day 2 is by version and device. The paywall headline test is postponed until two weeks after a stable release, because any result during the regression would mix two effects. Written in the decision journal: "Trial drop explained by 4.2 ANR on one device family; not a paywall signal."

Follow-up. On 4.2.1, the ANR rate for the affected device family returned to the 4.1 level and trial starts by version matched again. That is the reversal that turns "plausible" into "likely".

Do this now

1. **Build a release timeline** for the last 90 days: version, date, rollout percentage, what changed (including SDKs, server and remote config).
2. **Next to each release, add error evidence by version:** crash rate, ANR or freeze rate, failed purchases, failed sign-ins.
3. **Add one business metric by version:** trial starts, purchases or day-1 return rate from your own events.
4. **Look for a segment match:** a version, OS or device where errors *and* the business metric moved together.
5. **Set a rollout habit:** staged rollout on Play [S19], phased release on the App Store [S08], with a written "halt if" rule and a day-2 check by version.
6. **Postpone growth tests** that would overlap an unresolved regression.

Don't change anything yet when...

- You have an error spike but no version or device concentration — investigate before you rewrite.
- The business metric moved in all versions equally; the release is probably not the cause.
- A growth test is scheduled during an unresolved regression — postpone it.
- You are tempted to ship a fix and a new feature together; ship the fix alone so you can see its effect.
- Your only error evidence is a handful of reviews; they justify an investigation, not a conclusion.

Worksheet

Release regression triage — release, error evidence and business evidence by version and segment, with a halt-or-proceed rule and a fix-priority check.

Optional: Next Move uses public data only, so it cannot see your crash or error data; when public reviews hint at a technical problem, it can propose an investigation checklist, never a diagnosis. Indevaro is being built to connect release and error sources with business metrics in the full product; that capability is planned, not part of the free tool.

Sources

- S05 — Metric definitions, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/reference/metrics-definitions/>
- S08 — Release a version update in phases (App Store Connect Help) — <https://developer.apple.com/help/app-store-connect/update-your-app/release-a-version-update-in-phases/>
- S18 — Android vitals (Android Developers) — <https://developer.android.com/google/play/vitals>
- S19 — Staged rollouts, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/6346149>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=paying_users can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

7.

App competitor analysis: listings, reviews and positioning

In this chapter: how to read reviews as a biased sample rather than as a vote, how to count themes honestly, and how to treat competitors' claims as claims. You will turn an afternoon of reading into one positioning decision.

The problem

A useful app competitor analysis compares apps serving the same user task in the same store and market. Start with one or two relevant alternatives, record what their listings actually say, and look for a difference your current app can demonstrate. Public pages reveal positioning; they do not reveal a competitor's conversion rate or exact revenue.

Build a comparison you can act on

Question	Your app	Competitor	Evidence to record
Who is it for?	Write the audience your page addresses.	Write the audience their page addresses.	Storefront, language, date and the exact listing field.
What does it help someone do?	Name the main task.	Name the main task.	The visible claim, separated from your interpretation.
How is the promise demonstrated?	Inspect the first screenshots.	Inspect the same part of their listing.	What the screenshot shows; do not infer features from a slogan.
Why do reviewers choose it?	Tag a defined sample of recent reviews.	Use a comparable sampling rule.	Themes with counts and sample size, never a share of all users.
What can you truthfully say better?	Verify it in your released app.	Check whether their listing addresses it.	One message to test, plus what would disprove the idea.

Select competitors by the job people hire the app to do, not only by category rank. A large general-purpose app can be an alternative, but its audience, brand recognition and traffic may make its aggregate ratings a poor benchmark. If a feature or price cannot be verified from the available evidence, mark it unknown.

Reviews are the only place where strangers tell you, in their own words, what they expected and what they got. They are also a trap. One eloquent 1-star review can reset a roadmap. A competitor's listing that says "the most private journal" can make your own page feel weak. And a spreadsheet that says "18% of reviews mention sync" quietly turns into "18% of users have sync problems" by the time it reaches a planning meeting.

What is going on

Reviews are a sample, and a biased one

Who writes reviews is not random. People write when they are delighted or annoyed, when the app asks at a good or bad moment, and when they have time. Several platform mechanics shape the sample further:

- **When you ask.** On the App Store the system rating prompt can be shown up to three times in a 365-day period [S07]. On Google Play the in-app review dialog has a time-bound quota, and it may not appear if you call it more than once in a short period [S17]. Where you place the prompt changes who answers.
- **How ratings are summarized.** Apple's summary rating is per territory, and you can reset it with a new version — Apple recommends doing so sparingly, and a reset does not remove written reviews [S07]. The rating shown on Google Play is weighted towards more recent ratings [S15].
- **Language and country.** You read the reviews in languages you understand, for the countries you open. That is a filter too.

So a review sample tells you **what some users say**, not **how many users feel it**. Always record the sample: platform, countries, languages, date range, star ratings included, and how many reviews you read.

Count themes, then say what the count means

A theme count is useful for comparing themes *inside the same sample* — "sync complaints are more common than pricing complaints in the 60 reviews we read" — and for comparing the same theme *across periods with the same sampling method*. It is not a percentage of users. Write it as "11 of 60 reviews read (US and UK, English, 1 July–15 September)", not "18% of users".

Use a simple code sheet:

1. Read each review once and tag up to two themes: a problem ("export fails"), a wish ("wants dark mode"), a reason for choosing the app ("no account needed"), a comparison ("switched from X").
2. Add the version and star rating.
3. Only after tagging, count.

Reasons for choosing the app are gold for positioning: they are the words real users use for the value you deliver.

Competitor claims are claims

A competitor's listing is a statement of what they want you to believe. It is evidence of their **positioning**, not of their product's quality or their results. Three rules:

- Do not copy a claim you cannot prove for your own app. On Google Play, ranking claims such as "#1" are not allowed in the listing, and unattributed testimonials are not allowed in the description [S14].
- Do not treat a competitor's review count as a measure of their downloads or revenue. Nothing public tells you that.
- Look for **gaps**: something users in *your* reviews value that neither you nor the competitor say clearly on the page.

Stay inside the review rules

Never offer rewards for ratings or reviews, and never pressure or trick users into rating. Google Play's policy forbids incentivized ratings and manipulating ratings, reviews or installs [S16]; its in-app review guidance says not to ask opinion questions such as "Do you like the app?" before or while showing the rating prompt [S17]. Apple states that manipulating reviews or rankings, or using third-party services to do so, can lead to removal from the Apple Developer Program [S06]. Apple asks for concise replies without marketing language [S07]; Google asks that replies not solicit or promote [S15].

Worked example

Fictional example · E7 · Quillnest · tags: ios, reviews, small-sample

Situation. Quillnest is an iOS journaling app made by a solo developer. Goal: sharper positioning. A larger competitor recently changed its subtitle to lead with privacy. The developer wants to rewrite Quillnest's subtitle and first screenshot to say "the most private journal".

Sample. 60 most recent written reviews on the App Store, US and UK, English, 1 July–15 September, all star ratings.

Theme count (up to two tags per review, so the tag counts add up to more than 60):

Theme	Reviews	Typical wording
Reason: "export to PDF/Markdown is easy"	11	"finally I can get my entries out"
Reason: "no account needed"	8	"no sign-up, just write"
Problem: iCloud sync delay	7	"entries show up late on iPad"
Wish: templates	6	"prompts for mornings"
Mentions privacy explicitly	3	"private"
Other / single mentions	31	—

Observations. In this sample, export and "no account needed" are the most frequent reasons users give for choosing the app. Privacy is mentioned in 3 of 60. The current Quillnest listing mentions export only in the fourth screenshot and does not mention "no account" at all.

Hypotheses.

1. Users choose Quillnest for data ownership (export, no account) more than for privacy as such. *Supported in this sample; not a measure of all users.*
2. Leading with "the most private journal" would copy the competitor's position without evidence that Quillnest is more private. *A claim the developer cannot prove.*
3. The sync-delay reviews point to a technical issue that could hurt ratings. *Needs a check (chapter 6), independent of positioning.*

Decision: *Position on data ownership, as a test.* Do not use "most private". Test a new first screenshot that says "Your journal, yours to export — no account needed" — both true of the current version — as a product page optimization treatment against the current screenshot, with a minimum run set from chapter 4's planning rule

[S04]. That split test, not a custom page, is what can show whether the message converts better for the same visitors. Separately, use the same screenshot on a custom product page [S03] for the next community post (chapter 9); its numbers describe how that community responds, not whether the message beats the default. Open a separate investigation of the sync delay. Revisit the subtitle only after the product page optimization test ends.

What would change the decision. If a larger, more recent sample (for example the next 60 reviews after the sync fix) shows privacy as a frequent reason for choosing Quillnest, reconsider.

Do this now

1. **Define your sample** before reading: platform, countries, languages, date range, number of reviews.
2. **Tag each review** with up to two themes, its star rating and its version.
3. **Count and write each count with its sample** ("11 of 60 reviews read...").
4. **Collect the exact phrases** users use for why they chose the app.
5. **Read one competitor's listing** and write down its three main claims, marking each "provable for us", "not provable for us" or "not relevant".
6. **Pick one positioning idea** that is true for your current version and supported by your users' own words, and plan to test it (chapter 4) rather than replacing the default page.
7. **Separate technical themes** (crashes, sync, login) into chapter 6's triage.

Don't change anything yet when...

- Your sample is small, one-sided (only 1-star), from one country or from before a major version.
- The theme you want to act on is a single vivid review.
- The positioning would rest on a claim you cannot prove for your current version.
- The most frequent theme is a bug — fix or investigate that first.
- You have just reset your App Store summary rating or changed your review prompt; the sample is shifting.

Worksheet

[Review-theme sheet](#) — sample definition, tagging grid, theme counts with their sample, user phrases, competitor claims and one positioning decision.

Optional: if your goal is positioning, Next Move can compare your public listing with one competitor's and, from a bounded sample of public reviews, propose a positioning comparison or listing copy draft. It reports review shares as shares of the sample it read, and asks you to confirm any feature it would mention.

Sources

- S03 — Custom product pages (Apple Developer) — <https://developer.apple.com/app-store/custom-product-pages/>
- S04 — Product page optimization (Apple Developer) — <https://developer.apple.com/app-store/product-page-optimization/>

- S06 — App Review Guidelines (Apple Developer) — <https://developer.apple.com/app-store/review/guidelines/>
- S07 — Ratings, reviews, and responses (Apple Developer) — <https://developer.apple.com/app-store/ratings-and-reviews/>
- S14 — Metadata policy, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9898842>
- S15 — Ratings and reviews, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/138230>
- S16 — User ratings, reviews, and installs policy, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9898684>
- S17 — In-App Reviews API (Android Developers) — <https://developer.android.com/guide/playcore/in-app-review>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=positioning can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

8.

How to measure an app growth experiment

In this chapter: how to decide, before you change anything, what you will measure, for whom, for how long and against what — and how to reach an honest verdict afterwards, including "inconclusive". Correlation, causation and small samples explained in plain language, with every rule of thumb's assumptions stated.

The problem

You changed something. A number moved. It is very hard not to connect the two. And when the number moves the wrong way, it is just as hard not to find a reason why "it doesn't count".

Three different statements get mixed up here, and this chapter is about keeping them apart:

1. **The metric changed.** Downloads per week went from 400 to 460.
2. **The goal was reached.** We wanted 450 per week; we are above it.
3. **Our action caused it.** Without the new screenshots, we would still be at 400.

The first is an observation. The second is a comparison with a target you set. The third is a claim about a world you did not observe — the one where you did nothing — and it needs the most evidence.

What is going on

Five things to write down before you change anything

Baseline. What the metric looked like before, over a period long enough to show its normal range. One week is rarely enough; 8–12 comparable periods are a practical starting range, not a universal sample-size rule. Seasonality, autocorrelation or structural changes can make that range insufficient.

Exposure. Who actually experiences the change. A new screenshot only affects people who see the product page or the search result that shows it. A custom product page only affects visitors who arrive through its URL or targeting [S03] [S11]. A change in a release only affects users on that version. Measure the metric for the exposed group where you can; the whole-app total dilutes the effect.

Window. When you will look, and for how long. Decide in advance. If you keep looking until the number looks good and stop there, you will almost always find a "win" — that is chance, not the change.

Confounders. Anything else that could move the metric in the same window: holidays and seasons, releases, price changes, featuring, ads, press or community posts, competitor launches, store outages — and **definition changes**. For example, Google Play's page dates the primary unique-click metric to June 2026 and the store-listing report update to July 2026, replacing the older conversion-rate headline with click-through rate [S12]; a before-and-after comparison across that change compares different metrics. Play still offers the older store listing acquisitions in Statistics and exports [S12], so a continuous series is available if you look for it.

Verdict rules. What result counts as "worked", "did not work", "harmful" and "inconclusive". Write them before you look.

Correlation and causation, in plain words

Two things that move together are **correlated**. That can happen because one causes the other, because something else causes both (a holiday increases both downloads and reviews), or by chance.

You get closer to **causation** by comparison: the same kind of people, in the same period, with and without the change. That is what product page optimization on the App Store [S04] and store listing experiments on Google Play [S10] do — they split traffic at the same time, so seasons and news affect both sides equally. When you cannot split traffic, look for a **comparison group** that did not get the change but faces the same conditions: another country where you did not change the page, or the other platform. Be careful: iOS and Android users differ, prices and offers differ, and store algorithms differ. The other platform is a helpful hint, never a true control group.

Small samples: how much can a number move on its own?

A rough check for **counts** (downloads, purchases, reviews in a period):

- **Assumptions:** events happen independently of each other, the underlying rate is stable over the periods you compare, and the counts are comfortably above 20. These are Poisson/normal sanity approximations, not significance tests; use exact Poisson/binomial intervals or longer windows for very small counts, clustered events or materially varying rates.
- **Rule of thumb for one period:** a single period's count usually falls within about $\pm 2\sqrt{N}$ of its long-run average N , by chance alone.
- **Rule of thumb for two periods:** two periods with the same underlying rate can differ from each other by about $\pm 3\sqrt{N}$ (more precisely $\pm 2.8\sqrt{N}$), because both of them wobble.
- **Examples:** 400 downloads a week → a week between 360 and 440 is unremarkable, and two weeks 55 apart are unremarkable. 25 purchases a week → one week within ± 10 , two weeks up to about 14 apart. 9 purchases a week → below the range where these rules are even reasonable; use longer periods.
- **Better still:** compare against your own 8–12-period normal range when the periods are comparable; it contains your app's real wobble, including effects the rule of thumb ignores.

For **rates** (conversion, trial-to-paid), report the estimated difference and any interval or precision information available, then use the planning rule in chapter 4 or the store's own test statistics. Apple reports a confidence level for product page optimization and suggests waiting for at least 90% [S04]; Google Play lets you set the confidence level and the minimum detectable effect, and declares a draw when the difference is smaller than that effect [S10]. Read these as the stores' own summaries under their own assumptions. Even a confident

result does not tell you exactly how *large* the true effect is, or whether it will hold once the test ends and the season, traffic mix or competition changes. And a store's result covers only the metric it measures — usually downloads or clicks (Install, Open or Pre-register clicks in current Play reports), not paying users.

Regression to the mean

If you act right after an unusually bad week, the next week is likely to be better whatever you do — unusual weeks tend to be followed by more ordinary ones. This is why "we fixed it and it recovered" is weak evidence when the fix came right after a dip. A baseline of several periods helps you see whether the dip was unusual.

"Inconclusive" is a result

An inconclusive result means the prespecified decision threshold was not reached; nonsignificance alone does not show that the effect is small. Report the estimated effect and interval or precision information available, then state whether the planned power or precision was sufficient to exclude a meaningful effect. Only an interval that rules out your prespecified meaningful effect supports excluding that effect under the analysis assumptions. Planned power alone cannot do this after the result is known; a low-powered or repeatedly peeked test does not establish that the effect is small. Record it as such (chapter 11), and do not quietly relabel it as a win or a loss.

Worked example

Fictional example · E8 · Pantrio · tags: ios, android, conflicting-signals, inconclusive

Situation. Pantrio (the meal-planning app from chapter 5) replaced its iOS screenshots on 1 September with a new set focused on "plan a week in five minutes". The change was made directly, not as a split test. Android's listing was not changed. Goal: more downloads, and ultimately more paying users. Four weeks later, the developer wants to know whether it worked.

Measurement card (written, belatedly, on 29 September):

Item	Entry
Baseline	iOS and Android, US, 8 weeks before 1 September
Exposure	iOS visitors who saw search results or the product page; Android not exposed
Window	4 weeks from 1 September
Confounders	US school year starting (meal planning is seasonal); no releases or price changes; no ads
Verdict rules	Not set in advance — a weakness

Observations (4 weeks after vs 8-week baseline, weekly averages):

- iOS conversion rate (App Store Connect [S05]): up from about 5.1% to about 5.7%.
- iOS unique-device impressions: down about 9%.
- iOS total downloads: roughly flat (+2%).
- Android downloads (listing unchanged): up about 10%.

- New paying users, both platforms: flat, about 30 per week on each platform, within the range of the baseline weeks (for a weekly count of 30, the rule of thumb gives about ± 11 for a single week).

The signals conflict. Conversion on iOS rose, which fits the new screenshots working. But downloads did not rise, because fewer people saw the app. Android — with no change — rose more than iOS, which fits a seasonal lift in interest that iOS should also have received. If iOS had the same seasonal lift as Android, then the flat downloads could mean the new screenshots did *worse*; or the drop in iOS impressions may have hidden an improvement. Paying users did not move beyond normal variation on either platform.

Hypotheses.

1. The new screenshots improved iOS conversion. *Possible*; the conversion rise is consistent with it.
2. The conversion rise is a mix effect: fewer impressions from broad browse traffic (which converts less), leaving a larger share of high-intent search visitors. *Possible*; impressions fell mainly in browse (checked in the source view [S23]).
3. Seasonality lifted interest on both platforms, and iOS search visibility fell for unrelated reasons. *Possible*.

Decision: *Inconclusive — keep the new screenshots, claim nothing.* The change is cheap to keep and shows no sign of harm. The journal entry reads: "Metric changed: iOS CR up, downloads flat. Goal not reached. Causation: not established; seasonality and traffic mix confound." For the next listing change, the developer will use product page optimization [S04] so both versions face the same season, and will write the verdict rules first. Separately, the drop in iOS browse impressions goes to a discoverability check (chapter 3).

Do this now

1. **Pick one change you made in the last three months** and fill a measurement card for it, honestly, even if late.
2. **Check your baseline range:** list the metric for the 8–12 comparable periods before the change.
3. **Write down the exposed group** and whether you can measure it separately.
4. **List confounders** in the window, including definition changes in any console.
5. **Apply the counts rule** (with its assumptions) or the store's own test result.
6. **Write a verdict:** worked, did not work, harmful or inconclusive — and the three separate statements (metric changed / goal reached / caused by us).
7. **For your next change,** fill the card *before* you ship it.

Don't change anything yet when...

- You have not written the baseline, window and verdict rules.
- A known confounder (holiday, release, price change, featuring, campaign) falls inside the window and you cannot separate it.
- You are still inside the window of a previous change to the same step.
- Your counts are too small for the window you chose; lengthen the window before you start.
- The metric's definition changes during the window.

Worksheet

Measurement card — baseline, exposure, window, confounders, verdict rules, the counts check with its assumptions, and the three separate statements.

Optional: each move Next Move proposes includes a measurement instruction — what to watch, over what window, and its limits. Indevaro is being built to record outcomes against that instruction in the full product; today, this card is how you do it.

Sources

- S03 — Custom product pages (Apple Developer) — <https://developer.apple.com/app-store/custom-product-pages/>
- S04 — Product page optimization (Apple Developer) — <https://developer.apple.com/app-store/product-page-optimization/>
- S05 — Metric definitions, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/reference/metrics-definitions/>
- S10 — Store listing experiments, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/12053285>
- S11 — Custom store listings, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9867158>
- S12 — Acquisition reports, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9859173>
- S23 — Acquisition, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/acquisition/acquisition/>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=downloads can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

9.

How to market your app after launch

In this chapter: how to run small, honest distribution experiments in places where your future users already talk — Reddit and other forums, X, Facebook groups — without spamming anyone. You will learn to fit each community's rules, lead with useful work, disclose your interest, and measure the result well enough to decide whether to continue.

The problem

When store traffic is flat, "post it somewhere" is the obvious next idea. It often goes badly. A launch link dropped into five communities on the same day gets removed, downvoted or ignored, and the developer concludes that "Reddit doesn't work" or "X is dead" — from a sample of one badly fitted post per place.

The opposite mistake is quieter: spending weeks crafting content for a community without any way to tell whether a single user arrived because of it.

What is going on

Communities are not channels; they are places with rules

Each platform has platform-wide rules, and most communities add their own:

- **Reddit.** Reddit's rules say that each community's culture is shaped by rules its moderators enforce, ask you to abide by those rules, to participate where you have a personal interest, not to spam or manipulate content, and not to mislead others about who you are [S20]. Many subreddits add specific rules on self-promotion, link posts or designated promotion threads — read them before posting.
- **X.** X's authenticity policy prohibits bulk, duplicative, irrelevant or unsolicited content and inauthentic use of engagement features; automation must follow X's Developer Policy and may not be used aggressively to manipulate traffic or engagement [S21]. Mass-replying the same link under strangers' posts is exactly the kind of activity to avoid.
- **Facebook groups.** Read the current platform policy and each group's own rules, usually in the group's About section; the group's admins enforce those local rules.
- **Forums and Discord/Slack communities.** Usually have pinned rules and a promotion channel. The same principles apply.

If a community does not allow self-promotion, the honest options are: participate without links, ask a moderator, or choose another community. Evading a rule with a "friend's" account or a disguised post is both against platform rules [S20] [S21] and bad evidence — you would learn nothing you could repeat.

Lead with useful work

A post earns its place when a reader who never installs your app still leaves with something useful. Examples of useful work you can show:

- A **how-to** that solves the problem your app solves, done manually: "How I plan a 3-day hike with offline maps on any phone" (with a paper checklist).
- A **teardown of your own data or decision**: "We tested two first screenshots for three weeks; here is why the result was inconclusive".
- A **free resource**: a template, a checklist, a small tool that works without your app.
- An **honest build log**: what you shipped, what broke, what you learned.

The app link comes after the value, clearly labelled as yours.

Disclose your interest, every time

Say plainly that you made the app. "I built this" in the first or last line is enough. Disclosure is a Reddit rule in spirit (be authentic, don't mislead [S20]) and a trust issue everywhere else. Never post as a "happy user" of your own app, never ask friends to upvote or reply, and never send unsolicited direct messages to people who commented. X's rules prohibit bulk, aggressive, high-volume unsolicited replies, mentions or DMs, and prohibit automation or coordinated engagement used to manipulate traffic or metrics [S21]; compliant automation is governed by the Developer Policy, but avoid it for this experiment.

Make the experiment measurable before you post

Small communities send small numbers, so you need a clean way to see them:

- **App Store**: a custom product page has its own URL, and App Store Connect reports impressions, downloads and conversion per page [S03]. App Store Connect also shows web referrers and supports campaign links [S23]. Look under the right source: taps on a link inside another app — the Reddit or X app, for example — count as app referrers, and on iOS so do taps from non-Safari browsers such as Chrome; only Safari taps count as web referrers [S23]. A dedicated custom page per community lets you see that community's arrivals separately.
- **Google Play**: a custom store listing can be reached through its own URL [S11], and Play's acquisition report can break down visits by UTM source and campaign for ads-and-referrals traffic [S12]. (UTM tags are the `utm_source` and `utm_campaign` parameters you add to a link, for example `&utm_source=walkingforum`, so reports can tell which link a visit came from.)
- **Your site**: if you link to a landing page first, count clicks there — but be aware that each extra step loses people.

Decide in advance what would make the channel worth a second post: for example, "at least 20 unique-user Install Clicks from the custom listing within 7 days, and at least a few trials" — numbers chosen for *your* app and goal, not a benchmark. Use the exact metric the store reports: on Google Play choose Install Clicks, Open Clicks or Pre-registration Clicks; these are intent measures and may include returning users depending on the report configuration [S12]. On the App Store it is downloads per custom product page [S03]. With counts that small, one post tells you little; 20 is an editorial threshold, not a statistical validity threshold (chapter 8's count approximation is unreliable for very small counts). Plan for two or three posts in comparable communities before you judge a channel.

Worked example

Fictional example · E9 · Fernpath · tags: android, distribution, small-sample

Situation. Fernpath (the hiking app from chapter 3) wants more UK installs before winter. The developer decides to try two places: a UK hiking subreddit whose rules allow self-promotion only in a weekly thread, and a long-running UK walking forum with a "Gear and apps" board that allows each developer one post about their own app, with disclosure.

Preparation (about 90 minutes).

- Read both communities' rules and the last month of posts in the relevant thread/board.
- Wrote one useful piece: "Preparing offline maps for a winter walk: a 10-minute checklist that works with any app", including a printable list. The Fernpath link appears at the end, after "Disclosure: I build Fernpath, an offline hiking map app for Android. It does step 4 for you, but the checklist works without it."
- Created a Play custom store listing with a unique URL for each community [S11], with the same assets as the default listing, so any difference in clicks comes from the audience, not the page.
- Wrote a channel experiment card: continue if either community brings at least 20 unique-user Install Clicks in 7 days *and* the post is not removed; stop otherwise.

Observations after 7 days.

Community	Post status	Listing visitors (custom URL)	Unique-user Install Clicks	Comments
Subreddit weekly thread	Stayed up; low visibility in the thread	34	9	2
Walking forum	Stayed up; moderator pinned the checklist for the season	120	41	14 (mostly about the checklist)

Interpretation. The counts are small; the difference between the two is large enough to notice but it rests on one post each. The forum post's value came mostly from the checklist being pinned — a sign the content was useful to that community, which is repeatable. Nothing here says the app would do well in other forums.

Decision: *Continue with forums, pause the subreddit, repeat before scaling.* The walking forum allows one post per developer, and that post is used: in that forum, the developer only replies in the existing thread. The second useful piece (a different topic, same disclosure) goes to a different, comparable walking forum whose rules allow it, with its own custom listing URL; a third follows in another in four weeks. Do not post in the subreddit's weekly thread again until there is a genuinely new resource to share. No DMs to commenters; reply in public only. Judge the channel after the third forum post.

Do this now

1. **List five communities** where people with the problem your app solves already talk. For each, write the platform, the community's own rules on self-promotion, and whether links are allowed.
2. **Drop any community** where honest participation with a link is not allowed, unless you are willing to participate without a link.

3. **Write one piece of useful work** that stands on its own without the app.
4. **Write your disclosure line** and put it in the post, not in a comment.
5. **Create a measurable path:** one custom product page [S03] or custom store listing [S11] per community, or a link with UTM tags [S12].
6. **Fill the channel experiment card:** the continue/stop rule in numbers you chose, the date you will judge, and how many posts you will make before judging.
7. **Post by hand, once per community.** Reply to comments in public. No automation, no DMs, no vote requests.

Don't change anything yet when...

- A community's rules forbid your post; choose another community instead of rewording to slip past them.
- You cannot measure arrivals from the post separately; set up the custom page or link first.
- The post only works if readers install the app; make it useful on its own first.
- You have judged a channel on a single post; plan at least two or three comparable posts.
- A post was removed; read the removal reason and the rules before posting anywhere similar.
- Your store listing is mid-test (chapter 4); a traffic spike from a post can disturb the test's traffic mix.

Worksheet

Channel experiment card — community, rules, useful work, disclosure, tracking link, continue/stop rule, and results per post.

Optional: ready-to-adapt post drafts for Reddit and forums, X and Facebook groups are in this book's distribution kit. Each gives value before the link, discloses the author's product, and has a slot for the specific community's rules.

Sources

- S03 — Custom product pages (Apple Developer) — <https://developer.apple.com/app-store/custom-product-pages/>
- S11 — Custom store listings, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9867158>
- S12 — Acquisition reports, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9859173>
- S20 — Reddit Rules (Reddit Inc.) — <https://redditinc.com/policies/reddit-rules>
- S21 — Authenticity policy (X Help Center) — <https://help.x.com/en/rules-and-policies/authenticity>
- S23 — Acquisition, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/acquisition/acquisition/>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move?goal=downloads can prepare one move from public evidence for this chapter's goal. Nothing runs until you choose Generate, and this book works without it.

10.

How to prioritize growth work across your apps

In this chapter: how to compare possible pieces of work — across one app or several — by goal, effort, confidence and what you give up by choosing them, without inventing ROI figures. You will plan two hours of real work and leave a buffer.

The problem

Most indie developers do not have a shortage of ideas. They have a list: fix the sync bug, try new screenshots, reply to reviews, write a post for a forum, localize into German, add an annual plan, look into the drop on Android. With more than one app, the list is three times longer and every app feels neglected.

The usual ways of choosing are all flawed. **The newest idea** wins because it is exciting. **The loudest problem** wins because a review was rude. **The spreadsheet** wins because someone typed "ROI: 4.2x" into a cell with nothing behind it. And **the bigger app** always wins, even when the smaller one has the one cheap, decisive fix of the week.

What is going on

Compare on things you can actually state

For each candidate piece of work, write five things. None of them requires a forecast you cannot back up.

1. **Goal fit.** Which goal does this serve — downloads, paying users or positioning — and is that the goal you chose for this app this month? Work that serves no current goal goes to a "later" list.
2. **Evidence strength.** How many independent kinds of evidence point to this being worth doing (chapter 1)? Use a small scale: *none* (an idea), *one kind*, *two or more independent kinds*. Two dashboards reporting the same event count once.
3. **Effort.** A range in minutes or hours, including the boring parts: preparing assets, waiting for review, writing the measurement card. Label it an estimate.
4. **Reversibility.** How easy is it to undo? Promotional text on the App Store can be changed any time [S01]; a price change or an app rename cannot be cleanly undone.
5. **Cost of waiting.** What happens if this waits a week? A live regression hurts users every day (chapter 6). A listing test idea usually loses nothing by waiting.

Then add a sixth, the one people forget:

6. **Opportunity cost.** What will *not* happen if you do this? Two hours on a paywall redesign are two hours not spent confirming whether a bug is costing you trials.

Ordering rules that do not need ROI

Use these in order:

1. **Stop active harm first.** A confirmed regression, a policy problem, a broken purchase flow.
2. **Then the cheapest decisive check.** If 30 minutes of looking would settle which of two bigger jobs is right, do the 30 minutes first.
3. **Then a well-prepared test** of the best-supported idea for the current goal.
4. **Then maintenance** that protects the other work (review replies, a small fix that keeps a test clean).
5. **Leave a buffer.** Plan about 100 of the 120 minutes and keep about 20 as a buffer. Something will take longer. If nothing does, the buffer goes to the "later" list's cheapest item.

"Watch" is a valid entry. An app with nothing pressing gets a re-check date, not invented work.

Why no ROI column

A return-on-investment number needs a credible estimate of effect size, and for most indie apps there is none: small counts, few past tests, changing seasons (chapter 8). A made-up ROI makes the list *look* rigorous while hiding the real reasoning. When you do have a genuine estimate — from a finished test on the same app, for example — write it down with its source and range. Otherwise write "effect unknown" and let evidence strength, reversibility and cost of waiting do the work.

Worked example

Fictional example · E10 · Tidewell, Lanternly and Pocketloom · tags: portfolio, ios, android

Situation. A solo developer runs three small apps: **Tidewell** (tide times, iOS and Android, paid subscription), **Lanternly** (reading tracker, iOS, one-time purchase) and **Pocketloom** (knitting row counter, Android, free with ads). It is Saturday morning, and there are two hours.

Candidate list:

#	App	Work	Goal fit	Evidence	Effort (est.)	Reversible	Cost of waiting
1	Tidewell	Investigate failed subscription purchases reported after release 3.1	Paying users (current goal)	Two kinds: four reviews since release + purchase-failed events up on 3.1 only	30–45 min to confirm and scope	Yes (investigation)	High: every day of failures loses subscribers
2	Lanternly	New first screenshot test	Downloads (current goal)	One kind: review themes (chapter 7)	60–90 min to prepare; test runs weeks	Yes	Low

#	App	Work	Goal fit	Evidence	Effort (est.)	Reversible	Cost of waiting
3	Pocketloom	Localize listing into German	Downloads (not a current goal)	None beyond a hunch	2–3 hours	Yes	Low
4	Tidewell	Reply to eight unanswered reviews	Positioning / trust	—	20 min	Yes	Low–medium
5	Lanternly	Annual price change	Paying users (not this app's goal)	None	1 hour + weeks of measurement	Hard to undo	Low

Ordering.

1. **#1 comes first:** active harm, two kinds of evidence, cheap to confirm, high cost of waiting.
2. **#4 next:** short, protects Tidewell's page while the fix is in progress, and the replies can mention that a fix is coming — without marketing language [S07].
3. **#2 only if time remains:** preparing the test brief (chapter 4) is useful even if the test starts next week.
4. **#3 and #5 go to "later":** no evidence and no current goal; #5 is also hard to reverse.
5. **Plan:** #1 45 minutes, #4 20 minutes, #2 35 minutes — 100 in total, with a 20-minute buffer.

What happened. The investigation (#1) took 50 minutes, not 45: purchases failed on one OS version because of a changed StoreKit call. A fix was prepared for a phased release on Monday. Reviews (#4) took 20 minutes. The Lanternly test brief (#2) took 40 minutes instead of 35. The buffer absorbed both overruns and left 10 minutes for the journal.

Decision: *Two hours went to Tidewell's purchase failures and reviews, plus a started test brief for Lanternly.* Pocketloom got a re-check date and no work — a deliberate "watch", recorded as such. No ROI figures were used; the choice rested on active harm, evidence strength and cost of waiting.

Do this now

1. **List every candidate** you are considering, across all your apps, in one table.
2. **Fill goal fit, evidence, effort, reversibility and cost of waiting.** Leave ROI out unless you have a sourced estimate.
3. **Apply the ordering rules:** harm, decisive check, prepared test, maintenance.
4. **Plan about 100 minutes** of your 120 and keep about 20 as a buffer.
5. **Give every app without planned work a re-check date,** not guilt.
6. **After the session, write one line per item** in the decision journal (chapter 11): planned, actual time, result.

Don't change anything yet when...

- An item's only support is that it is new or exciting — park it on the "later" list with a note of what evidence would promote it.
- Two items depend on the same unknown — do the check that resolves it first.
- A hard-to-reverse change is competing with an unresolved regression.

- You are filling the buffer with new work before the planned work is done.

Worksheet

Two-hours portfolio sheet — candidate table, ordering rules, time plan with buffer, and "watch" entries with re-check dates.

Optional: Next Move works on one app per run and respects the time budget you set (two hours by default): it proposes a move sized to it, or a check. Comparing opportunities across a whole portfolio is something Indevaro is being built to do in the full product; for now this sheet does it by hand.

Sources

- S01 — Creating your product page (Apple Developer) — <https://developer.apple.com/app-store/product-page/>
- S07 — Ratings, reviews, and responses (Apple Developer) — <https://developer.apple.com/app-store/ratings-and-reviews/>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move can prepare one move from public evidence. Nothing runs until you choose Generate, and this book works without it.

11.

How to learn from app growth experiments

In this chapter: how to write down what you decided, what you actually did, and what happened — and then revise what you believe about your app and its users. A short decision journal turns scattered experiments into knowledge you can use in the next decision.

The problem

Six months into running an app, most developers can list what they changed. Very few can say what they learned. The screenshot test from spring? "I think it helped." The forum post? "Some people came." The price change? "Revenue went up, but so did downloads, so who knows."

Without a record, three things happen. You repeat tests you already ran. You keep beliefs that your own evidence contradicted. And your judgment never gets better, because you never compare what you expected with what you got.

What is going on

Four records, not one

A useful journal entry separates four things that are easy to blur:

1. **The decision.** What you chose, the evidence behind it, the alternatives you rejected and why, and what you expected to happen. Written *before* acting. This is your decision card (chapter 1), filed.
2. **The execution.** What you actually did, when, and how it differed from the plan. Plans change: the test ran at 50% instead of 30%, the post went out on a Tuesday instead of a Saturday, the fix shipped with an extra change. Record it.
3. **The outcome.** What the measurement card (chapter 8) showed, with the three statements kept separate: the metric changed or not; the goal was reached or not; the evidence for causation is strong, weak or absent. "Inconclusive" is an outcome.
4. **The learning.** What you now believe, how strongly, and what would change your mind.

Beliefs, stated so they can be wrong

A learning is a belief about your app and its users, written so that future evidence could contradict it.

Compare:

- *Vague*: "Users like simple screenshots."
- *Useful*: "For Coinmoth's US search visitors, a first screenshot about setup speed converts better than one about spending history. Confidence: moderate (one test, one market). Would revise if a test in another market shows no difference, or if the effect disappears after the new-year season."

Three habits keep beliefs honest:

- **Write the prior.** Before the outcome, write what you believed and how strongly. Afterwards, compare. If every result "confirms" what you thought, you are probably not writing priors.
- **Move in proportion to the evidence.** One inconclusive test should barely move a belief. A clean split test with a clear result should move it a lot. A before-and-after change during a holiday should move it very little.
- **Keep negative and null results.** "Custom page for Reddit: 9 visitors, no downloads" is valuable. It stops you, or a collaborator, from trying the same thing next quarter under a new name.

Review on a schedule

Once a month, read the last month's entries and ask:

- Which beliefs changed? Which should now change the default of what you do?
- Which decisions were right for the wrong reasons, or wrong for good reasons? (A good decision can have a bad outcome; judge the reasoning with the information available at the time.)
- Which open questions keep coming back? They are candidates for a proper test.

Keep it small

A journal nobody keeps is worse than a short one. One entry per decision, a few lines per section. A spreadsheet or a plain text file is enough.

Worked example

Fictional example · E11 · Coinmoth · tags: ios, learning, listing-test

Situation. Coinmoth (chapter 4) re-ran its product page optimization test with one treatment and 50% of traffic. This example shows the full journal entry — decision, execution, outcome and learning — and the belief revision that followed.

Decision (written 6 October):

- *Chose*: test "Set up a budget in 60 seconds" on the first two screenshots against the original, US, one treatment, 50% of traffic. Minimum run 7 full weeks (the planned ~9,600 visitors per arm); after that, apply if Apple reports the treatment better at its suggested confidence [S04]; stop at 90 days as inconclusive otherwise.
- *Evidence*: the previous test was inconclusive (too small); review themes mention quick setup; the treatment message is true of the current version.
- *Rejected*: the color-scheme variant (no written hypothesis).
- *Prior belief*: "Setup-speed messaging converts better for search visitors." Confidence: low.

- *Expected*: conversion up by about 0.8 points, if the idea is right; otherwise no difference.

Execution:

- Started 9 October, three days late, because App Review had questions about the new screenshots [S04].
- No other listing changes in the US during the test. A minor bug-fix release shipped on day 20 — noted as a possible confounder; it did not touch onboarding.

Outcome (measurement card, 27 November, day 49 — the end of the minimum run):

- The treatment was reported as performing better than the original at Apple's suggested confidence level [S04]. Observed conversion: about 4.1% original vs about 4.8% treatment.
- Metric changed: yes, for exposed visitors in the test. Goal (more US downloads): total US downloads rose about 7% over the test period, within a season that also lifts budgeting apps — so the goal was only partly reached, and the rise is not all attributable to the test.
- Causation: *reasonably strong for conversion* (simultaneous split test), *not established* for total downloads.

Decision taken on the outcome: apply the treatment as the default product page; keep the measurement card open for four more weeks to see whether the conversion difference holds outside the test.

Learning:

- *Updated belief*: "For Coinmoth's US search visitors, setup-speed messaging converts better than spending-history messaging." Confidence: moderate (one clean test, one market, one season).
- *Would revise if*: conversion falls back to the old level within four weeks after applying, or a UK test shows no difference.
- *Next use*: the same message goes into the custom product page for the next community post (chapter 9), as a separate, measured experiment.
- *Also learned*: the first test's "inconclusive" was a sample problem, not a message problem — recorded against the first entry, so it is not read later as "the message failed".

Decision: *Apply the tested page, keep watching for four weeks, raise confidence in the belief from low to moderate — not to high.*

Do this now

1. **Create a journal** — a spreadsheet or text file — with four sections per entry: decision, execution, outcome, learning.
2. **Back-fill your last three decisions** from memory and data, marking anything you are unsure of as "not recorded".
3. **For your next decision**, write the prior belief and the expected result before acting.
4. **When the measurement window closes**, write the outcome with the three separate statements.
5. **Write the learning as a belief** with a confidence level and a "would revise if".
6. **Put a monthly review** in your calendar: 30 minutes to read the month's entries.

Don't change anything yet when...

- You have not written down what you expect to happen — you will not be able to tell whether the outcome surprised you.
- The previous change to the same step has no recorded outcome; close that entry first.
- A single result would flip a belief that several earlier results supported; look for a confounder before revising strongly.
- The "learning" you want to record is really a hope ("it will probably work next time").

Worksheet

Decision journal — decision, execution, outcome (three statements) and learning (belief, confidence, would revise if), plus a monthly review page.

Optional: the full Indevaro product is being built to keep this record for you — linking a decision to what was done, the measured result and the lesson used in the next recommendation. Today, the free Next Move tool gives you a move and a measurement instruction to start an entry; the journal is yours.

Sources

- S04 — Product page optimization (Apple Developer) — <https://developer.apple.com/app-store/product-page-optimization/>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move can prepare one move from public evidence. Nothing runs until you choose Generate, and this book works without it.

12.

Your next two hours of app marketing

In this chapter: everything in this book, compressed into one worksheet you can complete in two hours, on paper, for one app. You finish with one move, a ready material, a way to measure it and a journal entry — or with a well-founded decision to wait. An optional shortcut through the free Next Move tool is described at the end; you do not need it.

The problem

Reading about decisions is not the same as making one. The most common outcome of a book like this is a list of good intentions and no change on Monday. The fix is a fixed, time-boxed routine you can repeat every week or every month: the same steps, the same sheet, the same two hours.

What is going on

The worksheet follows the book's chapters in order. Here is the route and roughly how long each step takes. The times add up to about 100 minutes, leaving about 20 as a buffer — the same rule as chapter 10.

Step	What you do	Chapter	Time
0	Choose one app, one platform, one market, one goal	1	5 min
1	Write the observation that worries you, with its definition	1, 2	10 min
2	Fill the five funnel steps for two comparable periods; mark blind spots	2	15 min
3	Name the first step whose rate changed — or "none"	2–6	5 min
4	Write three hypotheses, including a boring one; one check for each	1, 3–7	10 min
5	Choose a decision type — act, test, check or wait — and the single move	1, 10	10 min
6	Prepare the material for that move	3–7, 9	30 min
7	Fill the measurement card	8	10 min
8	Write the journal entry and a re-check date	11	5 min
—	Buffer	10	~20 min

The five materials

Step 6 asks you to produce something you can use immediately. Pick the one that matches your move:

1. **Listing copy draft** — a new subtitle, short description or first lines of the description, within the store's limits (for example 30 characters for an App Store subtitle, 80 for a Google Play short description) [S01] [S13], each claim true of the current version.
2. **Screenshot brief** — for each of the first screenshots: the message in one line, what the screen shows, and why a visitor from your main source cares. Not final artwork.
3. **Positioning comparison** — your three main claims next to one competitor's, marked "provable for us / not provable / not relevant", and the gap you will lead with (chapter 7).
4. **Investigation checklist** — the exact data to pull, in which console or tool, with the result that would confirm or rule out each hypothesis (chapters 2, 5, 6).
5. **Experiment card** — hypothesis, variant, audience, sample estimate, stop rule and pre-committed actions (chapter 4), or a channel experiment card (chapter 9).

When the right answer is "wait"

If step 3 finds no changed rate, or step 4's checks show the movement is within your normal range, the move is **wait** with a re-check date and a trigger. You still complete step 7 (what you will watch) and step 8 (the journal entry). That is a full, successful session.

Working with little data

If your app is new or tiny, step 2 will be mostly blind spots — App Store Connect, for example, hides some metrics until there are at least five first-time downloads [S05], and Play's peer benchmarks appear only when there is enough data [S12]. Then the best move is usually one of two things: a **check** that creates the missing data (add events, set up a custom page URL for tracking), or a **positioning** move that relies on qualitative evidence (reviews, competitor listings, community questions) and is cheap to reverse.

Worked example

Fictional example · E12 · Saltpine · tags: ios, small-sample, positioning, listing-test

Situation. Saltpine is a new iOS app for planning sea-kayak trips, launched six weeks ago by a solo developer. Around 60 downloads a week in the UK and the US combined, a one-time purchase to unlock route export. The developer has two hours on Sunday.

Step 0. App: Saltpine. Platform: iOS. Market: UK. Goal: more downloads (positioning first). Time: 120 minutes.

Step 1 — observation. "UK total downloads were 31 last week and 38 the week before (App Store Connect)." Definition checked: total downloads include redownloads [S05].

Step 2 — funnel. Impressions ~1,100/week; conversion ~3%; downloads 31–38; purchases 2–4 per week; no in-app analytics yet (blind spot for activation); purchase events only from App Store Connect.

Step 3 — changed step. None clearly. Two weekly counts around 35 can differ by about ± 17 by chance (chapter 8's two-period rule, $\pm 2.8\sqrt{35}$), so 31 vs 38 is ordinary — and both weeks sit inside the six-week range of 27–41.

Step 4 — hypotheses and checks.

1. Noise. *Check*: 6-week range was 27–41. → Supported.
2. The page does not say who the app is for. *Check*: read the listing as a stranger; compare with the two best-known competitors. → Both competitors lead with "routes and tides"; Saltpine's subtitle says "Plan your paddle" — generic.
3. Visibility is low for "sea kayak" terms. *Check*: source split. → 70% of impressions from search; conversion on search visitors not separable at this volume.

Step 5 — decision. Type: *test*, in the cheapest reversible form. Move: prepare a **screenshot brief** that leads with the one thing competitors do not say — offline tide-aware route export for sea kayaks — and test it on the default page as a product page optimization treatment against the current screenshots [S04]. The message difference is deliberately large, because Saltpine's traffic is small (step 7). Separately, prepare a **promotional-text draft** for a custom product page used in the next community post; Apple lists screenshots, promotional text and app previews (plus keywords) as what a custom page can vary [S03], so the subtitle stays as it is for now.

Step 6 — material (excerpt).

- Screenshot 1 message: "Plan a tide-safe route, then take it offline." Shows the route planner with the tide panel open.
- Promotional text draft for the custom page (170 characters max [S01]): "Plan sea-kayak routes around the tides, then export them to use offline on the water. Built for paddlers, not hikers." (117 characters) ✓
- Parked for later, on the default page with a new version: subtitle draft (30 characters max [S01]) "Offline kayak routes & tides" (28) — after two rejected drafts of 31 characters. A subtitle change needs a new app version [S01] and is not part of this test.

Step 7 — measurement. Product page optimization test, one treatment, 50% of UK traffic: about 550 impressions per arm per week. Chapter 4's planning rule for a move from 3% to 4% conversion ($p = 0.03$, $d = 0.01$) gives about 4,700 visitors per arm, so the minimum run is 9 full weeks — inside Apple's 90-day limit [S04]. After week 9, apply the treatment only if Apple reports it better at its suggested confidence; at 90 days without that, report the estimate and precision information available and record it as inconclusive unless the planned precision excludes a meaningful effect. This design is not powered to distinguish effects smaller than one point. The custom page's own downloads and conversion [S03] are recorded as a description of the community audience, not compared with the default page.

Step 8 — journal. Prior: "Specific positioning beats generic for this niche." Confidence: low. Re-check: after week 9 of the test.

Decision: *Test the specific positioning as a split test on the default page; keep the subtitle; use a custom page only to describe the community audience; add basic in-app events next session.* Time used: 1 h 50 min.

Do this now

1. **Print or copy the worksheet** below. Block two hours in your calendar.
2. **Complete it for one app, one market, one goal.** Do not switch apps halfway.
3. **Finish with a move, a material, a measurement card and a journal entry** — or with "wait" and a re-check date.
4. **Repeat on a fixed rhythm** (weekly or monthly), reading last session's journal entry first.

Don't change anything yet when...

- You cannot finish step 3 because every funnel step is a blind spot — your move is a check that creates data.
- Step 4's boring hypothesis (noise, season, a definition change) survives its check.
- The material in step 6 would need a claim you cannot prove for the current version.
- Another change to the same step is still inside its measurement window.

Worksheet

Your next two hours — the complete offline worksheet: steps 0–8, space for each answer, the five material templates and the time plan. It works on paper, without an account and without Indevaro.

Optional: a shortcut through Next Move

If you would rather start from a prepared draft, the free Next Move tool at indevaro.com/tools/next-move covers part of this worksheet using **public** evidence. You give it your public store listing (on the stores it supports), a market (country and language), a goal — downloads, paying users or positioning — a time budget (two hours unless you change it) and, for a positioning goal, one competitor. It returns one move with the public evidence behind it, a ready material (one of the five types above) and a measurement instruction; when public evidence is not enough, it returns a decisive check instead of a guess. It can draft steps 4–6 and suggest what to measure in step 7. It cannot see your private metrics, so steps 1–3 and the actual numbers in step 7 still need your consoles — and the journal in step 8 is still yours.

The Indevaro workspace connects store, revenue, analytics and error tools and keeps plans, drafts, tasks and follow-up together. The worksheet above is complete without assuming daily use or claiming outcome proof.

Sources

- S01 — Creating your product page (Apple Developer) — <https://developer.apple.com/app-store/product-page/>
- S03 — Custom product pages (Apple Developer) — <https://developer.apple.com/app-store/custom-product-pages/>
- S04 — Product page optimization (Apple Developer) — <https://developer.apple.com/app-store/product-page-optimization/>
- S05 — Metric definitions, App Store Connect Analytics (Apple Developer Help) — <https://developer.apple.com/help/app-store-connect-analytics/reference/metrics-definitions/>
- S12 — Acquisition reports, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9859173>
- S13 — Create and set up your app, Play Console Help — <https://support.google.com/googleplay/android-developer/answer/9859152>

Optional: try it on a real listing

The free Next Move form at indevaro.com/tools/next-move can prepare one move from public evidence. Nothing runs until you choose Generate, and this book works without it.

Worksheets

The twelve worksheets from the chapters, ready to print. You may print and copy them for your own use. Each one is also on the web at indevaro.com/playbook/app-growth.

Worksheet 1 – Decision card

Use with chapter 1, *Evidence before action*. One card per decision. Fill it before you change anything. Works on paper; no account needed.

App: _____ **Platform:** ☐ iOS ☐ Android ☐ both **Market (country · language):** _____

Date: _____

Goal this period: ☐ downloads ☐ paying users ☐ positioning ☐ other: _____

1. Observation (facts only – no "because")

Metric (exact name)	Value	Compared with	Period	Source (console / tool)	Definition checked?
					☐ yes
					☐ yes

Normal range of this metric over the last 8–12 comparable periods: from _____ to _____ → Is the new value outside it? ☐ yes ☐ no ☐ don't know

2. Hypotheses (at least three, one of them boring)

#	Possible explanation	Data that would weaken it	Checked? Result
1			
2			
3 (boring: noise / season / holiday / definition change)			

3. Evidence strength

- Independent kinds of evidence pointing the same way: ☐ none ☐ one ☐ two or more
- Any evidence pointing the other way? _____
- Known confounders in the period (release, price, featuring, ads, holiday, definition change):

4. Decision

- ☐ **Act** — the change: _____
- ☐ **Test** — the comparison (tool, audience, share): _____
- ☐ **Check** — the one missing piece of evidence and where to get it: _____
- ☐ **Wait** — nothing changes now.

Cost of being wrong: ☐ low, easy to reverse ☐ medium ☐ high, hard to reverse

5. Re-check and trigger

Re-check on: _____ I will act if: _____

I will drop this if: _____

6. Where this goes next

- ☐ Test brief (worksheet 4) ☐ Measurement card (worksheet 8) ☐ Decision journal (worksheet 11)

Reminder: "Wait" and "check" are complete decisions. Write the trigger so future-you knows what would change your mind.

Worksheet 2 – Missing-data map

Use with chapter 2, *Where growth breaks*. One sheet per app and market. Copy definitions from the platform's help pages, not from memory.

App: _____ **Platform:** ☐ iOS ☐ Android **Market:** _____ **Periods compared:** A = _____ B = _____
(same length, same weekdays if possible)

1. The five steps

Step	Metric you use	Source	Definition (short)	Period A	Period B	Rate from previous step A → B
Seen	e.g. impressions (App Store) / store listing visitors (Play)					—
Downloaded	e.g. first-time downloads, total downloads / install clicks					
Activated	your key action (first entry, first plan...)					
Paying	purchases, trials started, paying users					
Staying	renewals, day-7 return, repeat purchase					

First step whose rate changed: _____ ☐ none changed (look upstream: season, market, noise)

2. Definition-change check

- Did any console rename or redefine a metric in these periods? ☐ no ☐ yes: _____ (Example: Google Play replaced its store listing "conversion rate" with click-through rate in mid-2026; the older store listing acquisitions remain in Statistics and exports — use them for a continuous series.)
- Did your own analytics events change (renamed, new SDK, new app version)? ☐ no ☐ yes: _____

3. Blind spots

Step	What is missing	How to get it	Effort (est.)	Can the current decision wait for it?
				☐ yes ☐ no — decide with what I have and record the gap
				☐ yes ☐ no
				☐ yes ☐ no

4. Composition check

Did the mix of people change between A and B?

- First-time vs returning (redownloads): _____
- Country mix: _____

- Traffic source mix: _____

5. Next

- ☐ The changed step goes to chapter ___ ☐ First, close blind spot: _____ ☐ Wait; re-check on _____

Worksheet 3 – Discoverability source checklist

Use with chapter 3, *Discoverability*. One sheet per app and market, last 28 days.

App: _____ **Platform:** ☐ iOS ☐ Android **Market • language:** _____ **Period:** _____

1. Where visitors come from

App Store Connect → Acquisition → Sources (App Store search includes views and downloads from ads in search results)

Source type	Impressions	Share	Downloads	Conversion
App Store search				
App Store browse				
Web referrer				
App referrer				
Other				

Play Console → Store performance (Google Play *search* = searches for your app's name or brand; category searches such as "racing game" count under *explore*)

Traffic source	Store listing visitors	Share	Install clicks	Click-through rate
Google Play search (brand)				
Google Play explore (browse + category search)				
Ads and referrals				

Generic search/explore share: _____ % → ☐ keyword work is a strong candidate ☐ probably not my first lever (check listing and distribution first)

2. What each source can and cannot tell you

Source	Can tell you	Cannot tell you
Store console source split	Where visitors come from; conversion by source	How many people searched a term
Play search-term dimension	Terms that brought <i>your</i> visitors	Terms that did not bring you anyone
Your rank in a keyword tool	Where you appear for a term	Demand or traffic
Third-party popularity scores	Rough comparison between terms	Real search counts
Your search-ads data (if any)	Terms that got impressions/taps for your ads	Organic behavior

3. Top terms: demand × visibility × conversion

Term	Demand (source; mark "estimate")	My rank (market, date)	Visitors or conversion from it	Keep / test / drop

4. Rules check

- ☐ No competitor names, trademarks or irrelevant terms in keywords or text
- ☐ No ranking claims ("#1", "best of") in Play listing text or images
- ☐ No repeated keywords in name or description
- ☐ Every term describes something the app really does

5. One change

- ☐ Keyword swap ☐ Subtitle / short description test ☐ Custom product page / custom store listing for one audience ☐ No change — reason: _____

Test brief started? ☐ yes (worksheet 4)

Worksheet 4 – Test brief

Use with chapter 4, *Listing and conversion* (also for paywall and pricing tests from chapter 5). Write it before the test starts. Do not edit the verdict rules after you have seen results.

App: _____ **Platform:** ☐ iOS ☐ Android **Market:** _____ **Owner:** _____ **Start date:** _____

1. Message hypothesis

- **Who** (which visitors, from which source): _____
- **What they want to know:** _____
- **What the page tells them now:** _____
- **What we will show instead:** _____
- **Is every claim true of the current version?** ☐ yes ☐ no → stop

2. Test setup

- Tool: ☐ App Store product page optimization ☐ Play store listing experiment ☐ custom product page ☐ custom store listing ☐ other: _____
- Number of treatments/variants: ____ (fewer is better when traffic is small)
- Assets changed: ☐ icon ☐ screenshots ☐ preview/video ☐ feature graphic ☐ description text — ☐ tested as a bundle
- Share of traffic in the test: ____ % → each variant gets ____ %
- Needs App Review / new binary (e.g. alternate icons)? ☐ yes ☐ no

3. Can this test say anything? (planning estimate)

Assumptions: independent visitors; stable traffic and rate during the test; ~80% chance to detect a real difference of size d ; 5% false-alarm rate; one variant vs original. This is a rough normal approximation for equal, independent arms; use an exact or store estimator for low rates, small counts, unequal allocation, repeated users or devices, or clustered traffic. Not the test's own statistics.

- Current rate p = _____ Difference to detect d = _____
- Visitors per arm $\approx 16 \times p \times (1 - p) \div d^2$ = _____
- Visitors per arm per week: _____ → weeks needed $\approx$ _____
- Tool time limit (App Store PPO: 90 days; Play experiments: 6 months): _____
- ☐ Fits ☐ Doesn't fit → make the difference bigger, send more traffic, use the largest market, or record as a judgment call

4. Metric and stop rule

- Primary metric (exact definition): _____

- Guardrail metric (must not get worse, e.g. trials per download): _____
- **Minimum run** (planned sample per arm, rounded up to full weeks): _____ weeks — no decision before this, whatever the tool shows
- **Maximum run** (tool limit or my own, whichever is shorter): _____
- After the minimum run, stop when: ☐ the tool reports a result at the confidence I set (App Store: Apple suggests at least 90%) ☐ the maximum run is reached → record the estimate and interval or precision information available; do not infer a small effect from a nonsignificant result, and mark inconclusive unless the planned precision excludes a meaningful effect
- Frozen during the test: other listing changes in this market ☐, price ☐, releases touching onboarding ☐

5. Pre-committed actions

Outcome	What we will do
Treatment better	
Treatment worse	
Inconclusive / draw / "more data needed"	
Guardrail got worse	

6. After the test

Result: _____ Recorded in decision journal on _____ ☐

Worksheet 5 – Monetization diagnostic tree

Use with chapter 5, *Monetization*. Walk the steps in order and stop at the first branch that explains the change. Every branch is based on data you have; nothing here asks you to judge a paywall from a store page.

App: _____ **Platform:** ☐ iOS ☐ Android **Market:** _____ **Period compared:** _____ vs _____

The change you are explaining: new paying users / revenue went from _____ to _____

Step 0 – Is it outside the normal range?

Normal range (8–12 comparable periods): _____ to _____. Count-size check (only if $N > \sim 20$; assumes independent events and a stable rate; not a significance test): one period vs its long-run average $\pm 2\sqrt{N} \approx$ _____; two periods vs each other $\pm 2.8\sqrt{N} \approx$ _____.

- Inside the range → **Branch W: wait.** Re-check on _____.
- Outside → Step 1.

Step 1 – Did payment break?

Data: purchase-failed events, error logs, reviews mentioning payment, recent releases touching purchases.

- Errors up around a release → **Branch T: technical.** Go to worksheet 6.
- No change / no data → note "payment errors: _____" and go to Step 2.

Step 2 – Did acquisition change?

Data: first-time downloads/installs from the store console.

- New users fell about as much as paying users → **Branch A: acquisition.** Go to chapters 2–4.
- New users stable → Step 3.

Step 3 – Did the mix of new users change?

Data: first-time vs returning (redownloads), country mix, traffic source mix.

- Mix changed (e.g. many returning users who are not eligible for an introductory offer) → **Branch C: composition.** Explain the drop by the mix; consider a separate offer test for that group.
- Mix stable → Step 4.

Step 4 – Did activation change?

Data: your own event for the key action (first entry, first plan...).

- Share of new users reaching the key action fell → **Branch V: activation.** Look at onboarding and the last release.
- No event logged → **Branch ? : can't tell yet.** Add the event: _____. Stop here.

- Stable → Step 5.

Step 5 – Did paywall exposure or response change?

Data: "paywall shown" per activated user; "purchase/trial started" per paywall view.

- Fewer users see the offer → **Branch E: exposure** (placement, onboarding, a flag or config change).
- Same exposure, fewer start → **Branch P: persuasion**. Only now write a paywall/pricing test brief (worksheet 4).
- No "paywall shown" event → **Branch ? : can't tell yet**. Add it before any redesign.

Step 6 – Did trial-to-paid or renewal change?

Data: store subscription/cohort views, your purchase events.

- Trial-to-paid or renewals fell → **Branch R: retention/renewal**. Check billing issues, offer changes, and the value users reach during the trial.

Result

Branch reached: ☐ W ☐ T ☐ A ☐ C ☐ V ☐ E ☐ P ☐ R ☐ ? **Next action:** _____ **Re-check:**

Platform comparison note (hint only, not a control group): _____

Worksheet 6 – Release regression triage

Use with chapter 6, *Technical regressions*. Put release, error and business evidence on one timeline and decide whether to halt, fix first, or proceed.

App: _____ **Platform:** ☐ iOS ☐ Android **Release under review:** version _____ released _____

1. Release facts

- What changed (features, SDKs/dependencies, server, remote config): _____
- Rollout: ☐ App Store phased release (day ___) ☐ Play staged rollout at ___ % ☐ full release
- Other changes in the same window (price, listing, campaign): _____

2. Timeline by version

Day	Rollout %	Error metric (crash / ANR / failed purchase / failed sign-in) — new version	Same metric — previous version	Business metric (trials, purchases, day-1 return) — new	Business metric — previous
0					
1					
2					
3					

3. Segment match

- Where are the errors concentrated? OS version _____ device/model _____ country _____
- Does the business metric move in the same segment? ☐ yes ☐ no ☐ can't split
- Reviews describing the symptom since release: _____ (qualitative — supports an investigation, not a conclusion)
- Platform thresholds (Google Play Android vitals bad-behavior lines, overall/per phone model): crash 1.09% / 8%, ANR 0.47% / 8% — over? ☐ no ☐ yes. *These are enforcement lines, not targets.*

4. Assessment

- ☐ **Not a regression** — errors unchanged or not concentrated in the new version.
- ☐ **Plausible** — errors changed at release, concentrated in the new version, business metric moved in the same segment.
- ☐ **Likely** — plus: halting/rolling back or fixing reversed the business change in that segment.

5. Decision

- ☐ **Halt / pause rollout** (Play: halt staged rollout; App Store: pause phased release — max 30 days total)

- ☐ **Fix first** — fix release plan: version _____, rollout ____ %, day-2 check by version. On Play, if the broken release was halted mid-rollout, start the fix at no less than the halted percentage: a new release started before the previous rollout completed goes to the same group of users.
- ☐ **Investigate** — next check: _____
- ☐ **Proceed** — reason: _____

Growth tests to postpone until two weeks after a stable release: _____

6. Fix priority check

Question	Answer
Share of active users affected (estimate, source)	
Harm now (data loss, cannot pay, cannot open)	
Does it corrupt a running test or measurement?	
Could it affect store visibility (Play thresholds)?	
Effort to fix (estimate)	

→ If harm is real and the regression is at least plausible, the fix goes first in the two-hours sheet (worksheet 10).

Worksheet 7 – Review-theme sheet

Use with chapter 7, *Reviews and positioning*. Define the sample before you read. A theme count is a share of the reviews you read — never a share of users.

App: _____ **Platform:** ☐ iOS ☐ Android

1. Sample definition

- Countries: _____ Languages: _____
- Date range: _____ to _____ App versions covered: _____
- Star ratings included: ☐ all ☐ only _____
- Number of reviews read: _____ Selection: ☐ most recent ☐ other: _____
- Anything that shaped this sample (new rating prompt, rating reset, featuring, a release): _____

2. Tagging grid (up to two tags per review)

Tag types: **P** problem · **W** wish · **R** reason for choosing · **C** comparison with another app

#	Stars	Version	Tag 1	Tag 2	User's own words (short)
1					
2					
3					
...					

3. Theme counts (write each with its sample)

Theme	Type	Count	Written as
			"__ of __ reviews read (countries, dates)"

Technical themes (crash, sync, login, payment) → move to worksheet 6: _____

4. Users' own words for why they chose the app

1. _____
2. _____
3. _____

5. Competitor claims

Competitor: _____ Listing checked on: _____

Their claim	Provable for us?	Relevant to our users (per section 4)?
	☐ yes ☐ no	☐ yes ☐ no
	☐ yes ☐ no	☐ yes ☐ no
	☐ yes ☐ no	☐ yes ☐ no

Gap — something our users value that neither page says clearly: _____

6. Positioning decision

- Message we will test: _____ (true of current version ☐)
- Where: ☐ custom product page / custom store listing ☐ store test ☐ community post ☐ no change yet
- Would change our mind: _____

7. Rules reminder

- ☐ No incentives for ratings or reviews ☐ No opinion questions before the rating prompt ☐ Replies short, specific, no marketing ☐ No unattributed testimonials in the description

Worksheet 8 – Measurement card

Use with chapter 8, *Measuring an intervention*. Fill sections 1-5 **before** the change ships. Fill section 6 when the window closes.

App: _____ Change: _____ Ships on: _____

1. Baseline

Metric (exact definition and source): _____ Last 8-12 comparable periods: _____
 _____ → normal range _____ to _____

2. Exposure

Who experiences the change: _____ Can I measure the exposed group separately? ☐ yes, how: _____ ☐ no (the total will dilute the effect)

3. Window

From _____ to _____. I will not judge before _____.

4. Confounders in the window

- ☐ holiday/season ☐ release ☐ price change ☐ featuring ☐ ads ☐ community/press post ☐ competitor launch ☐ store outage ☐ metric definition change ☐ other: _____ Comparison group (if no split test): _____ — remember: the other platform is a hint, not a control group.

5. Verdict rules (write now, don't edit later)

Verdict	Rule
Worked	
Did not work	
Harmful	
Inconclusive	

Small-sample check

Counts: assumes independent events, a stable underlying rate and N above ~20. A sanity check, not a significance test.

- One period vs its long-run average N: $\pm 2\sqrt{N}$. N = _____ → $\pm$ _____
- Two periods vs each other: $\pm 2.8\sqrt{N}$. N = _____ → $\pm$ _____
- Better still: the 8-12-period normal range in section 1.

Rates: use the test brief's planning estimate or the store's own test result (its confidence level and minimum detectable effect, under its own assumptions).

6. Result (when the window closes)

- Observed: _____
- **The metric changed?** ☐ yes ☐ no ☐ within normal range
- **The goal was reached?** ☐ yes ☐ partly ☐ no
- **Our action caused it?** ☐ strong evidence (split test) ☐ weak (before/after with confounders) ☐ not established
- **Verdict:** ☐ worked ☐ did not work ☐ harmful ☐ inconclusive
- Copied to the decision journal on _____ ☐

Worksheet 9 – Channel experiment card

Use with chapter 9, *Distribution experiments*. One card per channel; one row per post. Post by hand. No automation, no DMs, no vote requests.

App: _____ **Goal:** ☐ downloads ☐ paying users ☐ positioning

1. Community

- Platform: ☐ Reddit ☐ forum ☐ X ☐ Facebook group ☐ Discord/Slack ☐ other: _____
- Community name and link: _____
- Its own rules on self-promotion (copy the relevant line): _____
- Links allowed? ☐ yes ☐ only in a promotion thread/day: _____ ☐ no → participate without a link, ask a moderator, or choose another community
- Platform-wide rules read? ☐ yes (date: _____)

2. Useful work

- What a reader gets even if they never install the app: _____
- Format: ☐ how-to ☐ teardown of our own data/decision ☐ free resource ☐ build log
- Disclosure line (in the post, not in a comment): " _____ "

3. Measurement path

- ☐ App Store custom product page URL: _____ ☐ Play custom store listing URL: _____ ☐ link with UTM tags (`utm_source` , `utm_campaign`) that Play can report for referral traffic: _____ ☐ landing page with click count

4. Continue/stop rule (numbers chosen for this app – not a benchmark)

- Continue if within ____ days: ____ install clicks (Play) or downloads (App Store custom page) from this path **and** the post stays up **and** _____
- Stop if: _____
- Number of comparable posts before judging the channel: ____ (at least 2–3)

5. Posts

#	Date	Post title / topic	Status (up / removed + reason)	Visitors	Installs	Trials/purchases	Public replies given
1							
2							
3							

6. Verdict

- ☐ Continue ☐ Change the useful-work format ☐ Try a comparable community ☐ Stop this channel Reason:
_____ Journal entry written ☐

Worksheet 10 – Two-hours portfolio sheet

Use with chapter 10, *A portfolio and two hours*. Works for one app or many. No ROI column unless you have a sourced estimate.

Date: _____ **Time available:** 120 min → plan about 100, keep about 20 as buffer

1. Candidates

#	App	Work	Goal fit (current goal?)	Evidence (none / one kind / two+)	Effort (est.)	Reversible?	Cost of waiting a week
1							
2							
3							
4							
5							
6							

2. Ordering rules (apply in order)

1. Stop active harm (regression, policy problem, broken purchase).
2. Cheapest decisive check (30 minutes that settles which bigger job is right).
3. A well-prepared test of the best-supported idea for the current goal.
4. Maintenance that protects other work.
5. Buffer.

Opportunity cost of my first choice (what won't happen): _____

3. Plan

Order	#	Planned minutes	Actual minutes	Result
1				
2				
3				
Buffer	—	20		

4. Watch list (apps or items with no work today)

App / item	Why it can wait	Re-check date	Trigger to act

5. Later list (no evidence or no current goal yet)

Item	Evidence that would promote it

After the session: one line per item in the decision journal ☐

Worksheet 11 – Decision journal

Use with chapter 11, *From action to learning*. One entry per decision. Keep it short. Copy this page as often as you need.

Entry

ID: _____ App: _____ Date decided: _____ Linked cards: decision card ☐ test brief ☐
measurement card ☐

A. Decision (before acting)

- Chose: _____
- Evidence: _____
- Rejected alternatives and why: _____
- **Prior belief:** _____ Confidence: ☐ low ☐ moderate ☐ high
- **Expected result:** _____

B. Execution (what actually happened)

- Started: _____ Finished: _____
- Differences from the plan: _____
- Other changes in the same period: _____

C. Outcome (when the measurement window closes)

- The metric changed? ☐ yes ☐ no ☐ within normal range — detail: _____
- The goal was reached? ☐ yes ☐ partly ☐ no
- Our action caused it? ☐ strong evidence ☐ weak ☐ not established
- Verdict: ☐ worked ☐ did not work ☐ harmful ☐ inconclusive
- Decision taken on the outcome: _____

D. Learning

- **Updated belief:** _____
- Confidence now: ☐ low ☐ moderate ☐ high (moved from _____ because _____)
- **Would revise if:** _____
- Next use of this learning: _____

Monthly review (30 minutes)

Month: _____ Entries reviewed: _____

Question	Notes
Which beliefs changed this month?	
Which should change what I do by default?	
Right for the wrong reasons / wrong for good reasons?	
Open questions that keep coming back (candidates for a proper test)	
Null or negative results worth remembering	

Worksheet 12 – Your next two hours

Use with chapter 12. The complete routine for one app, one market and one goal, in about two hours. It works on paper, offline, without an account and without Indevaro. Everything you need is on these pages; the chapter numbers point to the explanation if you want it.

Before you start: open your store console (App Store Connect and/or Play Console), any analytics or subscription tool you use, and your last journal entry. Set a timer.

Step 0 – Scope (5 min)

- App: _____
- Platform: ☐ iOS ☐ Android (pick one for this session)
- Market (country · language): _____
- Goal: ☐ more downloads ☐ more paying users ☐ clearer positioning
- Time budget: 120 min (or _____)
- Last session's re-check date and trigger (from the journal): _____ → triggered? ☐ yes ☐ no

Step 1 – Observation (10 min) · chapters 1–2

Write the number that worries you — or the number your goal depends on — as a fact:

Metric (exact name)	Value	Compared with	Period	Source	Definition, in one line

Did the definition or reporting of this metric change during the period? ☐ no ☐ yes → note it and treat the comparison with care.

Step 2 – Funnel for two comparable periods (15 min) · chapter 2

Step	Metric	Period A	Period B	Rate from previous step A → B	Blind spot?
Seen				—	☐
Downloaded					☐
Activated					☐
Paying					☐
Staying					☐

Normal range of the key count over the last 8–12 periods: _____ to _____

Small-count check (assumes independent events, a stable rate and N above ~20; a sanity check, not a significance test):

- One period vs its long-run average $N: \pm 2\sqrt{N} = \pm \underline{\hspace{2cm}}$
- Two periods vs each other: $\pm 2.8\sqrt{N} = \pm \underline{\hspace{2cm}}$

Step 3 – The step that changed (5 min)

- ☐ First changed rate is at step: _____
- ☐ Nothing is outside the normal range → likely **wait** (go to step 5, choose "wait")
- ☐ Can't tell — too many blind spots → likely **check** (go to step 5, choose "check")

Mix check — did the kind of people change (returning vs new, country, source)? _____

Step 4 – Hypotheses (10 min) · chapters 1, 3–7

#	Possible explanation	One check that would weaken it	Result of the check
1			
2			
3 (boring: noise / season / holiday / definition change)			

Anything technical (crashes, failed purchases, a recent release)? ☐ no ☐ yes → it goes first (chapter 6).

Step 5 – Decision (10 min) · chapters 1, 10

- ☐ **Act** ☐ **Test** ☐ **Check** ☐ **Wait**
- The single move, in one sentence: _____
- Evidence strength: ☐ one kind ☐ two or more independent kinds ☐ none (then it is a check or wait)
- Cost of being wrong: ☐ low / reversible ☐ high / hard to reverse
- Opportunity cost (what I'm not doing): _____

Stop and choose "wait" or "check" if: the change is inside the normal range · a known confounder overlaps · the move needs a claim that is not true of the current version · another change to the same step is still being measured.

Step 6 – Prepare the material (30 min) · chapters 3–7, 9

Tick one and fill it in.

☐ **A. Listing copy draft**

Store limits: App Store name and subtitle 30 characters each, keyword field 100, promotional text 170 · Google Play name 30, short description 80, full description 4,000.

What each tool can change: App Store custom product page — screenshots, promotional text, app previews (and its own keywords), not name or subtitle · App Store product page optimization — icon, screenshots, app previews · Google Play custom store listing — name, icon, descriptions, graphics · Google Play store listing experiment — icon, feature graphic, screenshots, descriptions.

Field	Current	Draft	Characters	Every claim true now?
				☐
				☐

☐ B. Screenshot brief

Position	Message (one line)	What the screen shows	Why the main visitor source cares
1			
2			
3			

☐ C. Positioning comparison

	Our app	Competitor: _____
Main claim 1		
Main claim 2		
Main claim 3		
Provable for us?		—
Users' own words (from reviews)		

Gap we will lead with: _____

☐ D. Investigation checklist

#	Data to pull	Where	Result that confirms	Result that rules out
1				
2				
3				

☐ E. Experiment card

- Hypothesis (who / what they want / what we show now / what we'll show): _____
- Tool and audience share: _____

- Planning estimate (rates): visitors per arm $\approx 16 \times p \times (1 - p) \div d^2 = \underline{\hspace{2cm}} \rightarrow \text{weeks} \approx \underline{\hspace{2cm}}$ (*assumes independent visitors, stable rate, ~80% chance to detect d, 5% false-alarm rate*)
- Minimum run (planned sample, rounded up to full weeks — no decision before): $\underline{\hspace{2cm}}$
- Stop rule after the minimum run: $\underline{\hspace{4cm}}$
- If better: $\underline{\hspace{2cm}}$ If worse: $\underline{\hspace{2cm}}$ If inconclusive: $\underline{\hspace{2cm}}$

Step 7 – Measurement (10 min) · chapter 8

- Metric and source: _____
- Exposed group: _____
- Window: _____ to _____ (no judging before _____)
- Confounders to watch: _____
- Verdict rules — worked: _____ · did not work: _____ · harmful: _____ · inconclusive: _____

Step 8 – Journal and re-check (5 min) · chapter 11

- Prior belief: _____ Confidence: ☐ low ☐ moderate ☐ high
- Expected result: _____
- Re-check date: _____ **Act if:** _____
- Entry copied to the decision journal ☐

Time check

Step	Planned	Actual
0-1	15	
2-3	20	
4-5	20	
6	30	
7-8	15	
Buffer	20	
Total	120	

Done. You have one move, a material, a way to measure it and a journal entry — or a deliberate "wait" with a trigger. Both are complete sessions.

Optional: the free Next Move tool can draft steps 4–6 and suggest what to measure in step 7 from your public store listing (on the stores it supports, with an optional competitor) for the market, goal and time budget you choose. It cannot see your private metrics, so steps 1–3, the actual numbers in step 7 and the journal remain

yours. You do not need it to complete this sheet.

Source ledger

Date checked for every row: **2026-10-23**. All rows were re-checked for this release; S10 and S18 use the current canonical URLs noted below, and S22 remains a blocked historical row. Update owner for every row:

Indevaro editorial. Re-verify all rows at least every 90 days and before each release (see [design/PLAYBOOK_EDITORIAL_BRIEF.md](#) §9).

Quote scope: the book paraphrases by default. "Paraphrase only" means no verbatim text is used. "Short quote" means at most one fragment under 15 words, in quotation marks, attributed.

Verification methods: **direct text** = current first-party page text read directly (including a documented canonical redirect); **browser** = page read in a browser (read only, no interaction); **search index (historical)** = page could not be read directly, so content was confirmed only through a search-engine snippet. A blocked row supports no active chapter claim and must be re-verified before it is used again.

S01 Creating your product page (Apple Developer)

<https://developer.apple.com/app-store/product-page/>

App name and subtitle up to 30 characters each; keywords field 100 characters total; up to 10 screenshots; promotional text up to 170 characters and can be updated without a new app version; description changes need a new version.

Chapters 1, 3, 4, 10, 12. Quote scope: Paraphrase only. Verified: direct text.

S02 App Store search (Apple Developer)

<https://developer.apple.com/app-store/search/>

Search results depend on text relevance (title, subtitle, keywords, primary category) and on user behavior (downloads, ratings and reviews, and more).

Chapters 3. Quote scope: Paraphrase only. Verified: direct text.

S03 Custom product pages (Apple Developer)

<https://developer.apple.com/app-store/custom-product-pages/>

Up to 70 additional product page versions; can vary screenshots, promotional text and app previews; can have their own keywords and unique URLs; usable in Apple Ads; App Store Connect reports impressions, downloads and conversion per page. (The page's own average conversion figures are deliberately not used.)

Chapters 3, 4, 7, 8, 9, 12. Quote scope: Paraphrase only. Verified: direct text.

S04 Product page optimization (Apple Developer)

<https://developer.apple.com/app-store/product-page-optimization/>

Tests of icon, screenshots and app previews; up to three treatments; a test runs up to 90 days; traffic proportion is chosen; Apple suggests waiting for a treatment to reach at least 90% confidence; new metadata needs App Review; alternate icons must be in the app binary.

Chapters 4, 7, 8, 11, 12. Quote scope: Short quote ("at least 90% confidence"). Verified: direct text.

S05 Metric definitions — App Store Connect Analytics (Apple Developer Help)

<https://developer.apple.com/help/app-store-connect-analytics/reference/metrics-definitions/>

Definitions of impressions (viewed > 1 second; includes product page views), product page views, conversion rate (total downloads and pre-orders ÷ unique-device impressions), first-time downloads, redownloads, paying users, proceeds, crashes; usage data based on users who opt in to share data; metrics shown only above minimum thresholds (for example at least five first-time downloads).

Chapters 1, 2, 5, 6, 8, 12. Quote scope: Paraphrase only. Verified: direct text.

S06 App Review Guidelines (Apple Developer)

<https://developer.apple.com/app-store/review/guidelines/>

2.3.3 screenshots should show the app in use; 2.3.7 accurate keywords and no stuffing with trademarks, competitor names or irrelevant phrases; digital content or features consumed in the app generally use In-App Purchase, while §3.1.1(a) permits specified external-purchase links subject to entitlement and storefront conditions, including a broader no-entitlement rule for buttons, external links and calls to action in US storefront apps; attempts to manipulate reviews or rankings can lead to removal from the Developer Program.

Chapters 3, 4, 5, 7. Quote scope: Paraphrase only. Verified: direct text.

S07 Ratings, reviews, and responses (Apple Developer)

<https://developer.apple.com/app-store/ratings-and-reviews/>

The system rating prompt can be shown up to three times in a 365-day period; summary rating is per territory and can be reset with a new version, which Apple recommends sparingly; resetting does not remove written reviews; replies should be concise and without marketing language.

Chapters 7, 10. Quote scope: Paraphrase only. Verified: direct text.

S08 Release a version update in phases (App Store Connect Help)

<https://developer.apple.com/help/app-store-connect/update-your-app/release-a-version-update-in-phases/>

Phased release to users with automatic updates over 7 days (1%, 2%, 5%, 10%, 20%, 50%, 100%); pause for up to 30 days in total; anyone can still update manually.

Chapters 6. Quote scope: Paraphrase only. Verified: direct text.

S09 Auto-renewable subscriptions (Apple Developer)

<https://developer.apple.com/app-store/subscriptions/>

Introductory offer types (free trial, pay as you go, pay up front); a customer can redeem one introductory offer per subscription group; developers can also create offer codes, promotional offers and win-back offers; App Store Connect has offer and cohort/retention views.

Chapters 2, 5. Quote scope: Paraphrase only. Verified: direct text.

S10 Run A/B tests on your store listing (Play Console Help)

<https://support.google.com/googleplay/android-developer/answer/12053285>

Experiments can be run on the default or custom Store Listings. Play distinguishes one default graphics experiment from up to five localized experiments. Default graphics experiments test the icon, feature graphic and screenshots; localized experiments can also test descriptions in up to five selected languages. Experiments support up to 2 variants; audience percentage is split equally across variants; minimum detectable effect and confidence level are set by the developer; a difference below the minimum detectable effect is a draw; "More data needed" result; experiments stop automatically after 6 months; Google recommends testing one asset at a time.

Chapters 3, 4, 8. Quote scope: Paraphrase only. Verified: direct text (canonical URL; old ledger URL inaccessible).

S11 Create custom store listings (Play Console Help)

<https://support.google.com/googleplay/android-developer/answer/9867158>

Up to 50 custom store listings; targeting by country/region, pre-registration, search keywords, custom audiences, a unique listing URL or a Google Ads campaign.

Chapters 3, 4, 8, 9. Quote scope: Paraphrase only. Verified: direct text.

S12 Understand and grow your app's user base — acquisition reports (Play Console Help)

<https://support.google.com/googleplay/android-developer/answer/9859173>

The page says that from June 2026 primary metrics are based on unique clicks and that from July 2026 store listing reports focus on intent clicks. Reports count unique users clicking Install, Open or Pre-register; click-through rate is the share of store-listing visitors who clicked one of those actions and replaces the legacy conversion rate. A store-listing visitor did not already have the app installed on any device; the default report includes new and returning users, and a user is counted once per day per dimension. Legacy store listing acquisitions remain available in Statistics and CSV exports. Traffic source "Google Play search" covers searches for the app's name or closely associated brand, while browsing, suggestions, charts and category searches count as "Google Play explore"; reports can be filtered by search term, country, language and (for ads and referrals) UTM source and campaign; peer benchmarks appear only when the peer set has enough data.

Chapters 2, 3, 8, 9, 12. Quote scope: Short quote ("replaces the legacy \"Conversion Rate\" metric"). Verified: direct text.

S13 Create and set up your app (Play Console Help)

<https://support.google.com/googleplay/android-developer/answer/9859152>

App name up to 30 characters, short description up to 80, full description up to 4,000; repetitive or irrelevant keywords can lead to suspension.

Chapters 3, 12. Quote scope: Paraphrase only. Verified: direct text.

S14 Metadata policy (Play Console Help)

<https://support.google.com/googleplay/android-developer/answer/9898842>

Title, icon and developer name must not indicate store performance or ranking, price or promotion; the policy's common-violation examples list such text or images (e.g. "#1", "App of the year", "10% off") in listing assets; no unattributed or anonymous user testimonials in the description.

Chapters 3, 4, 7. Quote scope: Paraphrase only. Verified: direct text.

S15 View and analyze your app's ratings and reviews (Play Console Help)

<https://support.google.com/googleplay/android-developer/answer/138230>

The public Play rating is weighted towards recent ratings; ratings can be broken down by country, language, app version, Android version and device; guidance for replies (e.g. don't solicit or promote).

Chapters 7. Quote scope: Paraphrase only. Verified: direct text.

S16 User ratings, reviews, and installs policy (Play Console Help)

<https://support.google.com/googleplay/android-developer/answer/9898684>

Do not manipulate ratings, reviews or installs; do not offer incentives for ratings; prompt in a clear, non-deceptive way.

Chapters 7. Quote scope: Paraphrase only. Verified: direct text.

S17 Google Play In-App Reviews API (Android Developers)

<https://developer.android.com/guide/playcore/in-app-review>

Review dialog has a time-bound quota (may not show if called more than once in a short period); do not ask opinion or predictive questions before or while showing it.

Chapters 7. Quote scope: Paraphrase only. Verified: direct text.

S18 Android vitals (Android Developers)

<https://developer.android.com/google/play/vitals>

Core vitals include user-perceived crash rate and ANR rate; bad-behavior thresholds are 1.09% (crash) and 0.47% (ANR) overall, with 8% per phone model; above a threshold Play may reduce visibility or show a warning on the listing; Play assesses the last 28 days and data comes from users who allow it. The per-phone-model threshold is an app/device-model aggregate, not a version-slice threshold.

Chapters 6. Quote scope: Paraphrase only. Verified: direct text (canonical URL after redirect).

S19 Release app updates with staged rollouts (Play Console Help)

<https://support.google.com/googleplay/android-developer/answer/6346149>

Staged rollout percentage is chosen and does not increase automatically; halting stops additional users receiving the update; users are chosen at random; a new release started before the previous rollout completed uses the same group of users, depending on the percentage.

Chapters 6. Quote scope: Paraphrase only. Verified: direct text.

S20 Reddit Rules (Reddit Inc.)

<https://redditinc.com/policies/reddit-rules>

Communities have their own rules enforced by moderators; Rule 2: abide by community rules, participate where you have a personal interest, do not spam or manipulate content; Rule 5: be authentic, do not mislead.

Chapters 9. Quote scope: Paraphrase only. Verified: direct text.

S21 Authenticity policy (X Help Center)

<https://help.x.com/en/rules-and-policies/authenticity>

No bulk, duplicative, irrelevant or unsolicited posting (content spam); no inauthentic use of engagement features (engagement spam); automation must follow X's Developer Policy and may not be used aggressively to manipulate traffic or engagement. (The former URL `/platform-manipulation` redirects here.)

Chapters 9. Quote scope: Paraphrase only. Verified: direct text.

S22 Spam — Community Standards (Meta Transparency Center)

<https://transparency.meta.com/policies/community-standards/spam/>

No active book claim. Direct page read returned HTTP 429 on 2026-10-23; the earlier generic statement is retained only as a blocked historical lead and must not be used until directly verified.

Chapters —. Quote scope: Paraphrase only. Verified: blocked (historical; HTTP 429; prior search-index snippet not re-verified).

S23 Acquisition — App Store Connect Analytics (Apple Developer Help)

<https://developer.apple.com/help/app-store-connect-analytics/acquisition/acquisition/>

Acquisition source types: App Store search (includes views and downloads from ads shown in search results), App Store browse, app referrer, web referrer, App Clip, institutional; on iOS, taps from non-Safari browsers and from links inside other apps count as app referrer; source data can be filtered by territory and device; custom marketing campaigns can be tracked.

Chapters 1, 3, 8, 9. Quote scope: Paraphrase only. Verified: direct text.

Deliberately not used

- Apple's average conversion figures on the custom product pages page: platform-wide averages across unlike apps; the book refuses conversion benchmarks.
- Android vitals memory thresholds and their future enforcement date: not needed for the book's decisions.
- Community pages that could not be read or confirmed: S22's Meta page returned HTTP 429 and remains a blocked historical row; group rules are not treated as platform-policy evidence.