

PLANNING YOUR MOBILE APP: AN HONEST PRE-LAUNCH CHECKLIST



Planning Your Mobile App: An Honest Pre-Launch Checklist

The most expensive mobile app decisions get made before anyone writes a line of code — in the gap between “we need an app” and a clear picture of who installs it, which phones it runs on, and who owns it at the end. This checklist closes that gap.

Work through it before you commission anything. The clearer your answers, the tighter your scope, and the fewer surprises later — and the easier it is for any honest development shop, ours included, to tell you what to build, what to skip, and when a mobile-friendly website would serve you better than an app at all. There are no prices in here on purpose: scope comes first, and cost follows scope.

1. Start With the Users and the Problem, Not the App

It’s tempting to open with an app idea. Start with the people and the problem instead — whether you even need an app falls out of that, and sometimes the answer is a website.

Answer First:

- Who are the users, and are they really on their phones when they use this?
- What specific problem does the app solve, in one sentence?
- What do people do today instead, and what does that cost in time or errors?
- Would a mobile-friendly website do the same job without an install?
- Will people install an app for this, and open it more than once?

The Rule:

An app your audience won’t install is worse than a website they’ll use. If nothing here needs the phone itself — offline use, the camera, GPS, notifications — a web app is often the smarter buy.

2. Choose Your Platforms: iPhone, Android, or Both

Every mobile decision starts with which phones you're building for. "Both" feels like the safe default, but it isn't always the right one — and it isn't free.

Decide Early:

- Which phones do your actual users carry — iPhone, Android, or a real mix?
- Is this a public app, or an internal tool for staff on company devices?
- Do you need both stores at launch, or can one platform go first?
- Do you need a tablet layout, or phone only?

Why It Matters:

For an internal app on company-issued phones, you may only need one platform. For a consumer app, you almost always need both — which makes the next question the important one.

3. Decide Native vs. Cross-Platform

Once you know you need both stores, the big cost lever is whether you build each platform separately (native) or share one codebase across both (cross-platform).

Weigh the Trade-Off:

- Native builds the best possible experience per platform — at roughly double the work.
- Cross-platform shares one codebase to both stores — faster and less expensive.
- Do you need cutting-edge performance or deep, platform-specific hardware features?
- How fast do you need to reach both audiences at once?

The Honest Answer:

For most business apps, cross-platform is the right call — modern frameworks are fast enough, and one codebase costs far less to build and maintain. Reserve native for the apps that genuinely need the last bit of performance or the deepest hardware access.

4. Map Offline and Device-Hardware Needs

The whole reason to install an app instead of visiting a website is what the phone can do. Name those needs early — they shape the entire build.

Inventory What the App Touches:

- Does it need to work with no signal, then sync when it's back online?
- Does it use the camera, GPS, or a barcode/QR scanner?
- Does it send push notifications, and what triggers them?
- Does it need Bluetooth, biometrics, or any connected hardware?

Why It Matters:

Offline sync and hardware access are the hardest, most valuable parts of a mobile build. If none of them apply, ask again whether you need an app or a website.

5. Plan the Backend and Data

The app on the phone is only half the system. Behind almost every app is a backend that stores data, logs users in, and connects to the systems you already run.

Trace the Data:

- Where does the app's data live, and who else needs to read or write it?
- How do users log in, and how are accounts and permissions managed?
- What existing systems must it connect to — CRM, payments, inventory, email?
- Does data flow one way, or in both directions, and how fresh must it be?
- What data is sensitive, and what does protecting it require?

The Rule:

Scope the backend at the same time as the app, not after. An app with no plan for where its data lives is half a project priced as a whole one.

6. Own Your Developer Accounts and Code

This is the mobile trap most companies never see coming. An app is published to the stores through developer accounts — and if those accounts aren't yours, neither is your app.

Get These In Writing:

- Is the app published under **your** Apple Developer and Google Play accounts, in your company's name?
- Who owns the source code and intellectual property when it's done?
- Who owns the app's data, and can you export it in a usable form?
- Can you change developers later without rebuilding or re-publishing from scratch?
- Who holds the signing keys, store credentials, and backend access?

The Bottom Line:

Many agencies publish your app under their own accounts, which quietly locks you in. Insist that the code, the data, and the store listings are yours — under your own Apple and Google accounts — before work starts.

7. Budget for OS Updates and Life After Launch

A mobile app is never "done." Apple and Google ship new operating system versions every year, tighten store rules, and change hardware — and an app that isn't maintained quietly stops working.

Plan For:

- Who updates the app as new iOS and Android versions ship each year?
- Who keeps it compliant with changing App Store and Google Play rules?
- How are crashes and bugs reported, triaged, and fixed?
- Who handles security patches and dependency updates?
- How do new features get requested and prioritized after launch?

Why It Matters:

An unmaintained app doesn't fail on launch day — it fails the next time the OS changes and no one is there to update it. Decide who keeps it healthy before you need them.

8. Define What Success Looks Like

If you can't tell whether the app worked, you can't tell whether it was worth building. Define success in measurable terms before you start.

Set Your Measures:

- What will be faster, cheaper, or possible that isn't today?
- How will you measure it — installs, active users, time saved, errors avoided?
- What does "good enough to launch" look like versus "everything we imagined"?
- Which features are must-have for launch, and which are nice-to-have later?
- When will you review whether it delivered?

The Bottom Line:

A must-have list and a nice-to-have list are the two most valuable documents in any mobile project. Write them before you ask for a proposal.

The Final Word

A mobile app is only as good as the thinking that goes in before the build. Answer these questions honestly and you'll walk into any conversation with a development shop knowing what you need — and able to tell whether they're solving your problem or selling you a project.

Next Step:

Fill in your answers to sections 1, 3, and 6 first — the users, native versus cross-platform, and ownership. Those three decide most of the scope. When you're ready to turn the checklist into a plan, QOS Software will scope it with you honestly, and tell you where a mobile-friendly website would serve you better than an app. Call **877.800.7672** or visit **qossoftware.com**.