TAPFLARE / RESEARCH & PRACTICAL GUIDANCE

How to Buy an App Prototype: Scope Matrix and Checklist

> Tapflare is a flat-rate monthly subscription for graphic design and website development. A dedicated project manager assigns each request to senior designers and developers, so companies get steady creative and web output without hiring.

Published by Tapflare · 2026-11-16 · how to buy an app prototype · app prototype · prototype vs mvp · app prototype cost · app prototype checklist · ui ux design services · figma prototype · scope matrix · design subscription · app design handoff

Original article: https://tapflare.com/articles/buying-an-app-prototype-screens-real-data



In this article

1. Executive Summary
2. Introduction and Background
3. Visual Fidelity, Working Behavior and Production Acceptance
4. Completing the Behavior Matrix
5. Scoping One Flow and Its Failure States
6. Choosing What to Simulate and What Must Be Real
7. Comparing Quotes and Choosing a Supplier
8. Defining the Next Handoff
9. Data Analysis and Evidence
10. Sample Scope Clause and Decision Worksheet
11. Implications and Future Directions
12. Frequently Asked Questions (FAQs)
13. Conclusion

Executive Summary

Buying an app prototype is a scoping decision before it is a design purchase. The useful question is which decision the demo must support: approving visual direction, testing a task with users, showing a real integration, or releasing a supported product. Each needs a different minimum deliverable, and the word "prototype" does not reveal which one a supplier is quoting. Existing glossaries already define the stages: one agency calls a prototype a sample version of the product and an MVP the most basic launchable version (www.future-processing.com), and Atlassian lists a clickable wireframe of a new app flow as a prototype example (www.atlassian.com). This report turns those labels into a behavior-by-deliverable scope matrix that buyer and supplier complete together.

The matrix marks each behavior (screens, navigation, validation and error states, search, stored data, accounts, external services, post-demo support) as absent, illustrated, simulated, connected to a test environment, or implemented for production with agreed evidence. Tool names do not settle the level. Figma documents prototype variables with conditional logic (help.figma.com), and its Make product can create a Payload-powered backend with a database, file storage and authentication (help.figma.com), so a clickable demo can be a picture, a simulation or a working app. GOV.UK guidance adds that code prototypes need not meet production standards (www.gov.uk).

Public prices show why scope, not a label, drives cost. INSAIM's page, updated 14 October 2026, lists MVP product design from $5,000 USD excluding VAT as a starting point (www.insaim.design) and states that it designs MVPs and does not build them (www.insaim.design). Flatstudio quotes MVP design from €15,000, a euro figure with a different scope (www.flatstudio.co), and Superside lists a $15,000 monthly minimum on an annual term (www.superside.com). These are single examples, not market averages, and they are not interchangeable with a recurring subscription's output.

Before any real-user release, expect separate security, privacy, accessibility and platform-review work. OWASP's verification standard is intended in part to support specifying security requirements in contracts (asvs.owasp.org), and Apple expects review submissions to be final versions (developer.apple.com). The practical advice: scope one realistic flow and its failure states, write acceptance examples a third party can reproduce, total one-time and recurring lines separately, confirm file and code access in writing, and name who designs, builds, verifies deployment and maintains.

Introduction and Background

Founders, SaaS operators and marketing or product leads regularly commission "a prototype" from a freelancer, a studio, a design subscription or a product team. The brief often fits on one line, and quotes can differ widely because each supplier imagines a different object: polished screens, a linked flow with error messages, or a working application with sign-in and stored records. All three can look identical in a demo.

This report is written for the buyer's side of that exchange. It treats commissioning as a specification problem: which behaviors must exist, at what level of reality, with what evidence, and who owns the next step. It does not teach development or assume every team needs every stage: a team approving visual direction can stop at illustrated screens, and a team that must process real payments needs a build supplier and a verification plan, not a bigger mockup.

Disclosure: this guide is published by Tapflare, which sells a flat-rate design and website development subscription managed by a dedicated project manager. Its described scope includes website and mobile-app mockups; it is not presented as an app-engineering, launch or maintenance service. Tapflare appears once in the supplier comparison for completeness, and no first-party claim is used as evidence. Other claims rest on independent pages fetched on 16 November 2026. Vendor prices and platform rules are point-in-time figures as read on that date; re-check them against the live pages before reusing them. Where no public benchmark exists, such as hours per screen or the share of prototypes that become MVPs, this report says so rather than inventing one.

Visual Fidelity, Working Behavior and Production Acceptance

Start with the question, not the label

Atlassian's comparison frames a prototype as testing how a product might look or function and an MVP as testing whether a usable product solves a real customer problem (www.atlassian.com). Hostinger's tutorial compresses it: a prototype tests usability and design, while an MVP tests real adoption and market demand (www.hostinger.com). Y Combinator's 2016 essay is blunter: MVP stands for minimum viable product, not minimum viable prototype, and real users and customers will use and pay for it (www.ycombinator.com).

These definitions describe stages, but buyers purchase deliverables. *Table 1* starts from the next decision and works backward to the smallest deliverable that could answer it.

Next decision	Question to answer	Minimum deliverable	Evidence to request	Not tested
Approve visual direction	Do look, hierarchy and tone fit the brand and audience?	Illustrated key screens	Source file access, component list, contrast checks	Task completion; data and logic
Test a task with users	Can a target user finish one realistic task, and where do they stall?	Linked flow with happy path, validation and error states	Test script, participant count, findings, list of simulated behaviors	Real data volume; security; integrations
Demonstrate a real integration	Does the external service behave as promised in our flow?	One behavior connected to a test environment	Named test environment, buyer-held test credentials, reproducible steps	Live-mode behavior; support obligations
Release a supported product	Can real users rely on it, and who fixes it?	Production implementation with acceptance evidence	Written acceptance results, deployment record, account owner, support terms	Anything not on the acceptance list

Table 1 shows that the same screens can sit in all four rows. What changes is the evidence and the named exclusions. A supplier who cannot fill the last column for a quoted deliverable has not yet scoped it.

Three axes: visuals, behavior and production acceptance

Nielsen Norman Group (NN/g) defines prototype fidelity as how closely it matches the look and feel of the final system, and says fidelity can be high or low in all or some of three areas: visuals, content and interactivity (www.nngroup.com). A prototype can be visually near-final while its behavior is a person playing the computer, which NN/g describes as screens presented to the user in real time by a person (www.nngroup.com).

A third axis is production acceptance. GOV.UK's Service Manual says prototype code does not need to meet the same standards as production code, says code prototypes can look and behave like the live site, and warns not to copy prototype code into production (www.gov.uk). The guidance is written for government services and is not a universal commercial rule, but the principle travels: realistic does not mean releasable.

Tools blur the axes. Figma's prototyping model is built on flows, the network of frames and connections on a page (help.figma.com), and variables hold values or saved states and support if/else conditions (help.figma.com). Separately, Figma documents a Make backend with a database, file storage, authentication and an admin panel (help.figma.com), and Make apps can be published as a website with a dedicated URL (help.figma.com). Nothing in that documentation shows that every Figma deliverable is static, and nothing shows that one is production-ready. Capability of a tool is not scope of a supplier.

What existing glossaries already cover

Top-ranking explainers define the stages. Uptech calls a prototype an early attempt to visualize a working solution and an MVP the bare form of the product (www.uptech.team). These are agency blogs, so the definitions are attributed claims, and they give no method for checking a quote. The matrix below fills that gap without requiring stages in sequence: the Google Ventures design sprint describes finishing a prototype in one day to learn without building and launching (gv.com).

Completing the Behavior Matrix

The matrix has eight rows and five levels. Table 2 shows a column for each level except Absent, which is recorded beside a row when nothing is provided. Before signing, the buyer marks the level required for the decision, the supplier marks the level included in the price, and any gap becomes a scope change or a named exclusion. The levels are:

- **Absent:** not provided and not implied by the demo.
- **Illustrated:** shown in a static frame, with no response to input.
- **Simulated:** responds to input along scripted paths, variables or fixed sample content, with nothing kept beyond the session.
- **Connected to a test environment:** calls a real service or database isolated from live data and live money.
- **Implemented for production with agreed evidence:** runs on live data and users, with written acceptance results.

Table 2 applies the levels to a fictional booking flow. It is a template: each cell describes what that level would look like, and the agreed level for each row should be recorded beside it.

Behavior	Illustrated	Simulated	Connected to a test environment	Production, with agreed evidence
Screens	Static frames of booking form and confirmation	Frames linked in a flow with scripted transitions	Coded front end reading test content	Shipped UI with accessibility results against the agreed WCAG level
Navigation	Menu drawn on each screen	Tappable menu and back behavior on agreed paths	Routes resolve in a running build	Deep links and back behavior verified on named devices
Validation and errors	One drawn error message	Variables trigger an error when a field is empty	Test service returns real error codes	Every agreed error path has a written pass or fail
Search	Drawn results list	Typed query returns a fixed set	Query runs on a test index with sample records	Relevance, empty-result and latency accepted on live data
Stored data	None, or labeled sample content	Session-only values, lost on reload	Records written to an isolated test database	Live database with backups, access policy and named owner
Accounts	Sign-in screen drawn	Any credentials "work"	Sign-in against a test authentication service	Real identity, password reset and permissions reviewed
External services	Service logo on a screen	Faked response such as "payment succeeded"	Test-mode calls, such as a payment sandbox	Live mode with owner-held keys and monitoring
Post-demo support	Handover note	File access and a walkthrough	Environment guide and test credentials transferred	Named maintainer, response terms, change process

Reading across a row shows what a demo hides: a simulated sign-in accepts any password, which is useful for a task test and meaningless as proof of authentication. Post-demo support, the row most often left blank, is covered in the handoff section below.

Screens

Frame count alone is not an adequate quote. Five frames may be five states of one screen or five separate screens, and "up to 20 screens" does not say whether error, empty and loading versions count. Specify a screen by state inventory (see the worked example), name viewport sizes, and say who supplies copy and imagery. Checks that can enter acceptance include contrast of at least 4.5:1 for normal text under WCAG 2.2 criterion 1.4.3 (www.w3.org) and pointer targets of at least 24 by 24 CSS pixels under 2.5.8 (www.w3.org). WCAG 2.2 is a W3C Recommendation dated 12 December 2024 (www.w3.org); a native-app buyer should ask which platform guideline applies instead.

Navigation

A flow that works only when the presenter clicks the agreed path is simulated; Figma's sharing model reflects this, since a link can open a prototype at a flow's starting point (help.figma.com). Figma also documents that unsupported properties fall back to a default dissolve in smart animate (help.figma.com), so demo animation may differ from what a build can do. Ask for the list of wired paths and mark every other tap target illustrated.

Validation and error states

These states turn a vague scope into testable lines. WCAG 3.3.1 requires that a detected input error is identified and described in text (www.w3.org). Criterion 3.3.3 requires correction suggestions when known, unless that would jeopardize security or purpose (www.w3.org), and 3.3.4 requires a mechanism for reviewing, confirming and correcting information before finalizing certain submissions (www.w3.org). The GOV.UK Design System says to show an error message next to the field and in an error summary, and to describe what happened and how to fix it (design-system.service.gov.uk). NN/g adds that error text should be explicit, human-readable, polite, precise and constructive (www.nngroup.com). A line such as "empty, malformed and already-registered email each show distinct text messages" can be checked by a reviewer.

Search

A drawn results list is illustrated. A fixed result set for one typed query is simulated. Real search needs an index and records, which is a data and build decision. Specify the empty-result state too: Material Design's legacy documentation recommends starter content (m2.material.io), and NN/g advises using empty states to communicate system status (www.nngroup.com).

Stored data

Stored data is where "prototype" and "MVP" most often collide. NN/g notes that a data-heavy prototype may need only a small realistic set, for example 50 realistic-looking rows, to support the tested tasks (www.nngroup.com), which simulates volume. A database that persists records is a different deliverable with an owner, a backup question and an access-control question. Supabase's documentation says a table without

Row Level Security is readable and writable by any role with a grant on it, and that once it is enabled no data is accessible through the API until policies exist (supabase.com). Whether a supplier includes that work is a scope question that a tool name cannot answer.

Accounts

Authentication is a service, not a screen. Supabase describes its Auth product as a way to implement authentication and authorization in an app (supabase.com), and Figma's backend page lists authentication among what a Make backend can create (help.figma.com). Ask whether sign-in is drawn, accepts any credentials, runs against a test identity service or protects real accounts.

External services

Third-party services often offer a safe middle level. Stripe says its sandboxes let users test an integration without making actual charges (docs.stripe.com), that test card numbers are valid only in sandboxes (docs.stripe.com), and that general sandboxes isolate settings and data from live mode (docs.stripe.com). Firebase's Local Emulator Suite likewise supports integration testing or QA without touching production data (firebase.google.com). These sources show that "connected to a test environment" is a documentable level. Live mode, which processes real payments (docs.stripe.com), is a separate step with separate key handling (docs.stripe.com).

Scoping One Flow and Its Failure States

Hypothetical Example: a fictional studio-booking onboarding flow

(Hypothetical Example.) A fictional company, Fernwood Studio Booking, wants an app where a new member creates an account, picks a class and books a slot. The goal is to test whether first-time users finish a booking, so the decision is "test a task with users", and the minimum deliverable is a linked flow with failure states, not a connected build.

The screen by state inventory is the working document. Each row is a screen, each column a state, and each cell is marked drawn, simulated or not required:

- **Sign-up:** default, missing email, malformed email, password too short, email already registered, loading.
- **Class list:** default, loading, empty (no classes that day), error (cannot load).
- **Slot picker:** default, slot full, slot just taken, no connection.
- **Confirmation:** default, double-tap prevention, back after confirming.
- **Account home:** upcoming booking present, no bookings yet.

Alternate paths belong in the same document: the user abandons mid-flow and returns, changes class after choosing a slot, cancels, or arrives with no network. A supplier can then quote against 5 screens and the listed states rather than "5 screens", and the buyer can see what is missing. For waiting states, NN/g's response-time limits are the reference: 0.1 second feels instantaneous, 1.0 second keeps the user's flow of thought, and

10 seconds is about the limit for keeping attention (www.nngroup.com). Apple adds that a blank screen can be read as a problem with the app (developer.apple.com), and suggests cached or placeholder data with a nonintrusive label when the connection is lost (developer.apple.com).

Writing acceptance as reproducible examples

Acceptance should be an example another person can repeat. Cucumber's Gherkin reference describes Given steps as setting the initial context of the system (cucumber.io), which suggests a simple format: Given the user has an account and class X has one slot left, When two users tap Book in the same second, Then the second sees the "slot just taken" message with a way to choose another time. Atlassian describes acceptance criteria as clear, testable conditions that determine whether work is done (www.atlassian.com), and the Scrum Guide says work is not part of an Increment unless it meets the Definition of Done (scrumguides.org). For a simulated flow, write what is scripted so no one assumes real concurrency handling.

Choosing What to Simulate and What Must Be Real

Simulate what the decision does not depend on

GOV.UK says prototypes let a team explore design ideas much faster and at lower risk than using production code (www.gov.uk), and its alpha phase centers on identifying and testing the riskiest assumptions (www.gov.uk). The rule that follows is to make real only the behavior that carries the riskiest assumption. If the risk is whether users understand the slot picker, simulate everything else. If the risk is whether a payment provider's confirmation behaves as expected, a test-environment connection is justified. NN/g warns that a polished prototype's polish might stop people improving it (www.nngroup.com), yet also that testers should be able to click through and use the system as they would the real product (www.nngroup.com). Polish should be proportionate, and interaction sufficient for the task.

Real data versus mock data

Mock data usually suffices for visual approval and task testing, and real data adds privacy obligations. The FTC's business guidance includes a "scale down" principle: keep only what you need (www.ftc.gov). The UK ICO says to integrate data protection from the design stage and process only the personal information necessary for the specific purpose (ico.org.uk), and GDPR Article 25(2) says that by default only personal data necessary for each specific purpose is processed (gdpr-info.eu). This is generic context, not legal advice; consult a qualified professional about personal data. Practically, use invented records and protect access, since GOV.UK requires prototypes to be password protected (www.gov.uk).

What a user study adds

A prototype answers usability questions only if someone tests it. NN/g recommends testing early with a simple prototype, before the design is implemented (www.nngroup.com). Its well-known guidance is no more than 5 users per round with several small tests, and its model says a first study with five participants finds 85% of usability problems (www.nngroup.com). Independent work disagrees about reliability: a peer-reviewed study of 60 users found random sets of five finding between 55% and 99% of problems, and sets of ten raising the

worst case to 80% (pubmed.ncbi.nlm.nih.gov). NN/g says quantitative studies need at least 20 users (www.nngroup.com). Specify participants, rounds and who recruits them; "user tested" alone is not a deliverable.

Reviews required before real users

A realistic prototype is not cleared for the public. GOV.UK says alpha services should not be available for the public to use (www.gov.uk) and that beta is where the best idea is built for real (www.gov.uk). Before release to real users, budget for work that no design deliverable contains:

- **Security:** OWASP's Application Security Verification Standard provides a basis for testing web application technical security controls (asvs.owasp.org), and OWASP's mobile project describes MASVS as a security standard for mobile apps (owasp.org). NIST's Secure Software Development Framework recommends a core set of high-level practices (csrc.nist.gov).
- **Privacy:** for example, Apple requires a privacy policy link in App Store Connect metadata and in the app (developer.apple.com).
- **Accessibility:** for example, status messages that can be programmatically determined (www.w3.org).
- **Performance:** for web builds, web.dev's Core Web Vitals thresholds are LCP within 2.5 seconds, INP of 200 milliseconds or less and CLS of 0.1 or less (web.dev), judged at the 75th percentile of page loads (web.dev).
- **Store requirements:** Apple expects final versions with functional URLs (developer.apple.com), asks for demo account information with the back end turned on (developer.apple.com), and says apps submitted for TestFlight beta should be intended for public distribution (developer.apple.com).

Comparing Quotes and Choosing a Supplier

Supplier models and public price signals

Table 3 groups supplier types with prices and scope statements from each supplier's own current page, read on 16 November 2026. It is not a market average, prices are shown in the currency each supplier lists (euros and US dollars are not converted, so they are not directly comparable), no supplier is assessed for quality, and a monthly subscription, a one-off design package and a build engagement are different purchases.

Supplier model	Public price signal	Stated scope signals	Questions to ask
Project design studio (design only)	INSAIM: MVP product design from $5,000 USD excluding VAT, updated 14 October 2026 (www.insaim.design)	UX/UI ready for development (www.insaim.design); building is excluded	What states and flows are inside the starting price? Who supplies research and copy?
Design studio with a build partner	Flatstudio: MVP design from €15,000, a euro figure (www.flatstudio.co)	Engineering partner contracted directly at zero markup (www.flatstudio.co)	Who signs with the build partner, and who verifies deployment?
Enterprise creative subscription	Superside: $15,000 monthly minimum, annual term (www.superside.com)	Pricing text read did not address UX or prototype scope	Is app UI included, and in what form?

Supplier model	Public price signal	Stated scope signals	Questions to ask
Mid-market subscriptions	Flocksy: from $1,199 per month, annual (www.flocksy.com)	App or prototype scope not stated on the page read	Is a multi-state flow one request or several?
Subscriptions listing app or UI work	Kimp lists website, app and UI designs (www.kimp.io); Designjoy treats Webflow development as a design request (www.designjoy.co)	Kimp states you own the designs and receive source files (www.kimp.io)	Is "app design" a mockup, a flow or a build?
Graphics-centered subscriptions	Penji lists front-end development on its top plan (penji.co); ManyPixels includes source files (www.manypixels.co)	App, UX and prototype scope not stated on the pages read	Is interface flow design a supported request type?
Tapflare (publisher of this guide)	Flat monthly plans; see the publisher's current pricing page, not quoted here	Graphic design, website and mobile-app mockups, and website development; native-app build, launch and maintenance are not described as included	Is the needed behavior level within mockup scope, and who builds beyond that?
Marketplace freelancer	Set per listing	Fiverr's help page says IP rights are granted on delivery of the order, subject to full payment (help.fiverr.com)	Is a written assignment part of the order?

Table 3 shows why price comparison fails here. Two suppliers can both say "app design" and mean a drawn frame and a flow with states respectively, and several pages read for this report did not state prototype scope at all. That silence is not a deficiency, but it is a reason to ask. A project designer selling defined deliverables may suit a one-off prototype better than a recurring plan, while a subscription may suit a team expecting many small design requests over months.

Comparing quotes on the same lines

Put every quote into the same line structure before comparing totals. Ask each supplier to state:

- **Screens and states:** the inventory, with counts per state type.
- **Flows and links:** which paths are wired and which targets are illustrated.
- **Supplied inputs:** who supplies copy, imagery, sample data and research.
- **Revisions:** how many rounds, and what counts as a new request.
- **Evidence:** acceptance examples and test notes.
- **Access:** file and project access, export formats, and exported code if promised.
- **Exclusions:** development, security review, accessibility audit, store submission, hosting and support.

Then total the supplier-entered lines. One-time fees (design, research, build, launch) are summed separately from recurring fees (hosting, support, subscription), and first-year cost is one-time plus twelve times the monthly recurring. (Hypothetical Example: arithmetic only, not prices.) If a supplier enters design 4,000, research 1,000, build 0 and launch 500, the one-time total is 5,500. If it enters hosting 50 and support 200 per month, recurring is 250 per month, or 3,000 over twelve months, and first-year cost is 8,500. A quote with a

zero build line is not cheaper than one with a build: it excludes it, and the exclusions list records that. No hours-per-screen or prototype-to-MVP conversion figure is used, because no public benchmark with a traceable originator was found.

Hiring and ownership

Treat the contract as the source of truth. The US Copyright Office says a commissioned work is a work made for hire only if all stated requirements are met (www.copyright.gov), and that it cannot give legal advice on a work's status (www.copyright.gov). Under the statute, copyright vests initially in the author (copyright.gov), and a transfer is not valid without a signed instrument (copyright.gov). In the UK, GOV.UK says a person commissioned to create a work will usually own the copyright, and advises agreeing ownership and licensing in writing before work begins (www.gov.uk). Jurisdictions differ, so a lawyer should review the clause.

Marketplace terms show what to look for. Upwork's optional sample terms say that where payment covers only part of the delivery, the assignment covers only the portion delivered and paid for (www.upwork.com), and require a bill of materials for background technology and third-party materials (www.upwork.com), while noting they are a sample that may not suit every contract (www.upwork.com). Any supplier can be asked for the same: assignment on payment, a materials list and source file access.

Defining the Next Handoff

A demo ends, and the next steps need owners. Write four names into the agreement: who designs, who builds, who verifies deployment and who maintains the result. They may be one party or four.

Files, access and exported code

Figma's documentation shows why access terms matter. Prototype-only links are meant for presenting to clients, stakeholders or usability testing (help.figma.com). Viewing and downloading Make code needs a Full or Dev seat (help.figma.com), publishing a Make file requires a Full seat on eligible paid plans (help.figma.com), and the Full seat covers Design, Make and Dev Mode (help.figma.com). Dev Mode shows autogenerated code snippets per layer (help.figma.com), and Figma's marketing page says developers can review and adjust code generated in Make (www.figma.com). These describe what tools allow; a buyer should ask who holds the seat, who owns the project and in what format files and exported code arrive. Figma's FAQ also notes that changes to layers in Figma Design are not applied back to an existing Make file (help.figma.com), which matters if designers and app builders edit separately.

Who builds, and where accounts live

Design and build suppliers are often different companies: Flatstudio describes a separate engineering partner contracted directly (www.flatstudio.co). When a build starts, the buyer should own the developer and store accounts. Apple's Developer Program lists 99 USD per membership year and requires an organization to hold a D-U-N-S Number, with government entities excepted (developer.apple.com). Google Play lists a one-time registration fee of US$25 (support.google.com) and requires new personal accounts to run a closed test with at least 12 testers opted in for 14 continuous days before production access (support.google.com). Google's internal testing track distributes to up to 100 testers (support.google.com), and TestFlight allows up to 10,000

external testers once the first build has been approved by App Review for TestFlight (developer.apple.com). These are platform fees and rules as of 16 November 2026, not supplier fees, and they belong in the quote worksheet.

Verification and maintenance

Deployment should be verified by someone other than the party who built the release, or by the buyer through a checklist. The US General Services Administration's revised guide for de-risking technology projects adds a section on vendor management (www.gsa.gov), a reminder that handoffs are managed, not assumed. Maintenance is a recurring cost: list hosting, monitoring, dependency updates and support separately, since a supplier has not agreed to what the quote omits. The US Bureau of Labor Statistics describes digital designers as people who develop, create and test interface layout, functions and navigation for usability (www.bls.gov), which is the design side of the boundary.

Data Analysis and Evidence

Table 4 collects the quantitative anchors used above. None is a market average, and the sample is not representative of all suppliers. Values are as read on 16 November 2026.

Measure	Value	Originator and scope
Enterprise subscription minimum	$15,000 per month, annual term	Superside's own page (www.superside.com)
Subscription entry price	From $1,199 per month, annual	Flocksy's own page (www.flocksy.com)
Apple Developer Program fee	99 USD per membership year	Apple enrollment page (developer.apple.com)
Google Play registration fee	US$25 one-time	Google Play Console help (support.google.com)
Problems found by 5 users	85% (model); 55% to 99% across sets (study)	NN/g; Faulkner study (www.nngroup.com) (pubmed.ncbi.nlm.nih.gov)

Together with Table 3, the figures show wide variation in packaging: a design-only starting point, a euro-priced studio engagement and a recurring enterprise minimum are different purchase types, and starting prices cannot be divided into hours or set against a subscription's hours. The platform fees show that release costs sit outside any design quote. Hostinger says investment can easily double when moving from design to development (www.hostinger.com), but gives no figure or originator, so it is an unquantified vendor claim used in no calculation.

Sample Scope Clause and Decision Worksheet

Sample scope clause (adapt with legal advice)

"The Supplier will deliver a clickable prototype of the [flow name] flow comprising the screens and states in Schedule A. Each behavior in Schedule B is delivered at the level marked: illustrated, simulated, connected to a test environment, or implemented for production. Behaviors marked simulated use scripted paths and sample content and do not store or process real user data. The Supplier will give the Client's named account project and source file access, an export in [formats] and a third-party materials list. Acceptance is a pass result for each example in Schedule C, reproduced by the Client. Development, security review, accessibility audit, store submission, hosting and support are excluded unless listed. Ownership of deliverables transfers to the Client on payment under a signed written assignment." A lawyer should review this text, since ownership rules differ by jurisdiction.

Decision worksheet (blank)

- **Decision this demo supports:** ____ (approve direction / test a task / show an integration / release a product)
- **Flow in scope:** ____ happy path, ____ failure states, ____ alternate paths
- **Levels required (buyer) / quoted (supplier), per matrix row:** ____
- **Data environment and owner:** ____
- **Supplied by whom:** research ____ copy ____ imagery ____ sample data ____; **revisions:** ____ rounds
- **Acceptance examples (Given/When/Then):** ____
- **One-time lines:** design ____ research ____ build ____ launch ____ total ____
- **Recurring lines (monthly):** hosting ____ support ____ subscription ____ total ____ x 12 = ____
- **File, project and code access:** ____
- **Next-step owners:** designs ____ builds ____ verifies deployment ____ maintains ____
- **Reviews before real users:** security ____ privacy ____ accessibility ____ performance ____ store ____

Implications and Future Directions

The distinction between a clickable demo and a working application is likely to keep narrowing at the tool level. Figma documents prototype logic, publishing as a website, and code download for seat holders (help.figma.com). As such tools spread, a demo's appearance will say less about what sits behind it, and buyers will need to ask which behaviors are simulated, which touch a test service and which handle live data. The same backend page states a current limit, key-value stores rather than a full SQL database (help.figma.com), an example of a documented constraint that can change.

For suppliers, a scope matrix is a sales aid as much as a control: a package that states what is illustrated, simulated, test-connected and production-ready gives buyers a basis for comparison, while a silent page invites questions. For buyers, a subscription, a project designer and a product team are not ranked options. A subscription may suit many small design requests, a project designer a one-off prototype, and a product-

development provider a supported application. Revisit the matrix at each decision. The main uncertainty remains each supplier's actual scope and handover: neither the word "prototype" nor a tool name establishes readiness for real users.

Frequently Asked Questions (FAQs)

How do I buy an app prototype?

Name the decision, scope one flow with its failure states, complete the behavior matrix with each supplier, compare quotes line by line, then confirm access, acceptance examples and next-step owners.

What is the difference between a prototype and an MVP?

Atlassian describes a prototype as testing how a product or feature might look or function, and an MVP as the simplest usable version that solves a core customer problem (www.atlassian.com). In purchasing terms, ask whether the deliverable runs on real data for real users with agreed acceptance and support. If not, it is not an MVP, whatever the label.

What should an app prototype include?

The screens and states needed for the flow under test, wired navigation along agreed paths, validation and error states, realistic sample content and a list of what is simulated. NN/g says fidelity varies across visuals, content and interactivity (www.nngroup.com), so the quote should say which are high.

How do I specify screens for a prototype?

Use a screen by state inventory, not a frame count: default, empty, loading and error states per screen, plus viewport sizes and who supplies copy.

Should a prototype use real data or mock data?

Use mock data unless the question depends on real data. The ICO and GDPR text favor processing only what is necessary for a specific purpose (ico.org.uk) (gdpr-info.eu), and NN/g notes that a small realistic set can suffice for testing (www.nngroup.com).

What does an app prototype cost?

No public market average with a traceable originator was found. Published examples appear in Table 3 and have different scope. Total each quote from its own one-time and recurring lines.

What happens after a prototype demo?

The team decides whether to stop, iterate, test with users, connect a test service or start a build. GOV.UK expects alpha code to be thrown away and treats beta as building for real (www.gov.uk). Name who designs, builds, verifies and maintains before the demo.

How do I hire someone to build an app prototype?

Define the decision and flow, request a written scope that fills the matrix, and agree ownership in a signed written assignment (copyright.gov), which a lawyer should review.

What checklist should I complete before development?

Complete the worksheet above.

Conclusion

Buying an app prototype is safest when treated as buying evidence for a specific decision. The label on a deliverable is the least reliable part of a quote, because visual polish, simulated behavior and production acceptance vary independently. The behavior matrix makes the differences visible: each behavior is marked absent, illustrated, simulated, connected to a test environment or implemented for production with agreed evidence, and buyer and supplier compare marks before work begins.

The rest follows from that discipline. Scope one flow and its failure states instead of counting frames, use mock data unless the question requires real data, write acceptance as reproducible examples, total one-time and recurring lines separately, and put access, ownership and next-step owners in writing. Budget separately for the security, privacy, accessibility, performance and store reviews that a release to real users involves. Not every team needs every stage, and the right supplier model depends on the decision at hand.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://www.future-processing.com/blog/mockup-vs-prototype-vs-mvp-whats-the-difference
2. https://www.atlassian.com/agile/product-management/minimum-viable-product
3. https://help.figma.com/hc/en-us/articles/14506587589399-Use-variables-in-prototypes
4. https://help.figma.com/hc/en-us/articles/32640822050199-Add-a-backend-to-a-functional-prototype-or-web-app
5. https://www.gov.uk/service-manual/design/making-prototypes
6. https://www.insaim.design/pricing
7. https://www.flatstudio.co/solutions/mvp-design-and-development
8. https://www.superside.com/pricing
9. https://asvs.owasp.org
10. https://developer.apple.com/app-store/review/guidelines
11. https://www.hostinger.com/tutorials/prototype-vs-mvp
12. https://www.ycombinator.com/blog/practical-design-mvp
13. https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity
14. https://help.figma.com/hc/en-us/articles/360040314193-Guide-to-prototyping-in-Figma
15. https://help.figma.com/hc/en-us/articles/31304412302231-Explore-Figma-Make
16. https://www.uptech.team/blog/mvp-vs-poc-vs-prototype

17. https://gv.com/sprint
18. https://www.w3.org/WAI/WCAG22/Understanding/contrast-minimum.html
19. https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html
20. https://www.w3.org/TR/WCAG22
21. https://help.figma.com/hc/en-us/articles/360039818874-Smart-animate-layers-between-frames
22. https://www.w3.org/WAI/WCAG22/Understanding/error-identification.html
23. https://www.w3.org/WAI/WCAG22/Understanding/error-suggestion.html
24. https://www.w3.org/WAI/WCAG22/Understanding/error-prevention-legal-financial-data.html
25. https://design-system.service.gov.uk/components/error-message
26. https://www.nngroup.com/articles/errors-forms-design-guidelines
27. https://m2.material.io/design/communication/empty-states.html
28. https://www.nngroup.com/articles/empty-state-interface-design
29. https://www.nngroup.com/articles/ai-data-prototype-testing
30. https://supabase.com/docs/guides/database/postgres/row-level-security
31. https://supabase.com/docs/guides/auth
32. https://docs.stripe.com/test-mode
33. https://firebase.google.com/docs/emulator-suite
34. https://www.nngroup.com/articles/response-times-3-important-limits
35. https://developer.apple.com/design/human-interface-guidelines/loading
36. https://developer.apple.com/design/human-interface-guidelines/alerts
37. https://cucumber.io/docs/gherkin/reference
38. https://www.atlassian.com/work-management/project-management/acceptance-criteria
39. https://scrumguides.org/scrum-guide.html
40. https://www.gov.uk/service-manual/phases/alpha
41. https://www.nngroup.com/articles/test-earlier-with-ai
42. https://www.ftc.gov/business-guidance/resources/protecting-personal-information-guide-business
43. https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/accountability-and-governance/guide-to-accountability-and-governance/data-protection-by-design-and-by-default
44. https://gdpr-info.eu/art-25-gdpr
45. https://www.nngroup.com/articles/usability-101-introduction-to-usability
46. https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users
47. https://pubmed.ncbi.nlm.nih.gov/14587545
48. https://www.nngroup.com/articles/how-many-test-users
49. https://www.gov.uk/service-manual/agile-delivery/how-the-beta-phase-works
50. https://owasp.org/www-project-mobile-application-security
51. https://csrc.nist.gov/pubs/sp/800/218/final
52. https://www.w3.org/WAI/WCAG22/Understanding/status-messages.html

53. https://web.dev/articles/vitals
54. https://www.flocksy.com/pricing
55. https://www.kimp.io/pricing
56. https://www.designjoy.co
57. https://penji.co/pricing
58. https://www.manypixels.co/pricing
59. https://help.fiverr.com/hc/en-us/articles/360011454477-FAQs-for-clients
60. https://www.copyright.gov/circs/circ30.pdf
61. https://copyright.gov/title17/92chap2.html
62. https://www.gov.uk/guidance/a-quick-guide-to-protecting-your-creative-work
63. https://www.upwork.com/legal
64. https://help.figma.com/hc/en-us/articles/360040531773-Share-files-and-prototypes
65. https://help.figma.com/hc/en-us/articles/33649966245783-Edit-the-code-of-a-functional-prototype-or-web-app
66. https://help.figma.com/hc/en-us/articles/31304586129559-Publish-update-or-unpublish-a-Figma-Make-file
67. https://help.figma.com/hc/en-us/articles/19813618057623-Dev-Mode-for-admins
68. https://help.figma.com/hc/en-us/articles/15023124644247-Guide-to-Dev-Mode-in-Figma
69. https://www.figma.com/solutions/ai-engineering-handoff-tool
70. https://help.figma.com/hc/en-us/articles/31722591905559-Figma-Make-FAQs
71. https://developer.apple.com/programs/enroll
72. https://support.google.com/googleplay/android-developer/answer/6112435
73. https://support.google.com/googleplay/android-developer/answer/14151465
74. https://support.google.com/googleplay/android-developer/answer/9845334
75. https://developer.apple.com/testflight
76. https://www.gsa.gov/index.php/blog/2024/09/11/a-revised-and-expanded-guide-for-derisking-government-technology-projects
77. https://www.bls.gov/ooh/computer-and-information-technology/web-developers.htm

THE PUBLISHER

About Tapflare

Tapflare gives companies an on-demand design and web team for a flat monthly fee.

How it works

Each subscriber has a dedicated project manager who learns the brand, assigns every request to a senior specialist in that kind of work, and reviews the result before it is delivered. Requests are submitted and discussed in the Tapflare portal, each plan includes a daily allotment of design or development time, and the client keeps full ownership of the export and source files.

What it covers

Graphic design, custom illustration, presentation design, logos and brand guides, video editing and motion graphics, website and mobile-app mockups, and website development, from design mockups to HTML, to drag-and-drop builders or custom code.

What it is not

Tapflare is not a marketing agency, a media-buying, SEO or copywriting service, a software product or a freelance marketplace. Where a reader needs one of those, the honest answer is to say so.

Visit

See Tapflare and its pricing.

Articles published here explain general practice and cite their sources. Prices and plan contents, including Tapflare's own, change; check the provider's current pricing page before deciding.

Website: https://tapflare.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.