

# Step-by-Step Cost Breakdown: What You Pay at Each Phase of App Development

*A practical guide for business owners and product teams | code-b.dev | 2026*

---

## Introduction

Most cost guides for mobile app development show a single number, \$50,000, \$150,000, or some broad range, without explaining where that number comes from. Business owners go into projects with budgets built on incomplete information, and the surprises start once the build is underway.

This guide takes a different approach. It walks through every phase of a mobile app project from start to finish, explaining what happens at each stage, what it costs, and what percentage of your total budget it should represent. By the end, you will know exactly what you are paying for and why, which puts you in a much stronger position when evaluating quotes, managing a project, or allocating a defined budget.

All figures are based on standard agency rates and real project experience across fintech, healthcare, logistics, and e-commerce. Where relevant, we have noted what drives costs to the higher or lower end of each range.

## The Six Phases at a Glance

Here is how a complete app development project breaks down across the six stages.

| Phase                           | Cost Range         | % of Budget | Timeline  | Priority |
|---------------------------------|--------------------|-------------|-----------|----------|
| <b>Discovery &amp; Planning</b> | \$3,000 – \$10,000 | 5–8%        | 1–3 weeks | Critical |
| <b>UI / UX Design</b>           | \$5,000 – \$20,000 | 8–15%       | 2–4 weeks | High     |

## Code B | Step-by-Step App Cost Breakdown

|                           |                      |        |            |          |
|---------------------------|----------------------|--------|------------|----------|
| <b>Development</b>        | \$20,000 – \$150,000 | 40–55% | 8–24 weeks | Core     |
| <b>QA &amp; Testing</b>   | \$5,000 – \$30,000   | 15–20% | 2–6 weeks  | High     |
| <b>Deployment</b>         | \$2,000 – \$8,000    | 3–6%   | 1–2 weeks  | Required |
| <b>Post-launch (yr 1)</b> | \$8,000 – \$40,000   | 15–25% | Ongoing    | Ongoing  |

Development is the largest single line item but also the most variable — backend complexity is the primary driver, not screen count. QA is consistently the most underbudgeted phase; cutting its costs more in production fixes than it saves upfront. Post-launch costs are real and recurring and need to be in the plan before launch, not treated as a separate decision afterwards.

## Phase 1: Discovery and Planning

Typical cost: \$3,000 – \$10,000 | 5–8% of total budget | 1–3 weeks

Discovery is where the project gets defined. It does not produce working software, which is why many clients are tempted to skip it. That is a mistake. Every ambiguity in scope that goes unresolved here gets resolved later, during development, where fixing it costs five to ten times more.

In practice, discovery means mapping user flows, defining the data structure, making key technical decisions, and documenting what is in scope and what is not. The output is a specification that the development team can build from without constant back-and-forth.

### What discovery does it cover?

| Deliverable                  | Cost Range      | What it covers                    |
|------------------------------|-----------------|-----------------------------------|
| <b>Requirements workshop</b> | \$800 – \$2,000 | Scope, user flows, business logic |

## Code B | Step-by-Step App Cost Breakdown

|                               |                   |   |
|-------------------------------|-------------------|---|
| <b>Technical architecture</b> | \$1,000 – \$3,000 | Database design, API structure, third-party decisions |
| <b>Wireframes</b>             | \$800 – \$2,500   | Low-fidelity screens and navigation map               |
| <b>Project plan</b>           | \$400 – \$1,000   | Timeline, milestones, delivery structure              |
| <b>Risk assessment</b>        | \$200 – \$500     | Identify unknowns before development begins           |

### What drives the cost up or down

A simple app with clear requirements costs \$3,000–\$5,000 in discovery. A product with multiple user roles, unclear business logic, or regulatory requirements takes longer to define and costs \$7,000–\$10,000. If requirements are well-documented before the engagement starts, cost is at the lower end. If the product idea is still evolving, it takes more time.

The return is straightforward: a \$5,000 discovery phase that catches one scope ambiguity consistently saves \$20,000–\$50,000 in rework downstream.

## Phase 2: UI/UX Design

Typical cost: \$5,000 – \$20,000 | 8–15% of total budget | 2–4 weeks

Design is not about making the app look attractive. It is about defining how users move through the product, what they see at each step, and how the interface communicates information without friction. Good design reduces support requests, improves retention, and makes development faster because the team builds from clear specifications rather than interpreting vague directions.

The design phase produces two outputs: UX design, which covers structure and flow, and UI design, which covers visual execution. Both matter, and conflating them is a common source of budget errors.

### Design cost breakdown

| Activity | Cost Range | Description |
|----------|------------|-------------|
|----------|------------|-------------|

## Code B | Step-by-Step App Cost Breakdown

|                                   |                    |   |  |
|-----------------------------------|--------------------|---|--|
| <b>Wireframes (high-fidelity)</b> | \$1,500<br>\$5,000 | – | Clickable prototype, full navigation, content layout |
| <b>Visual / UI design</b>         | \$2,000<br>\$8,000 | – | Brand-aligned screens, typography, colour system     |
| <b>Design system</b>              | \$1,000<br>\$3,500 | – | Component library for developers to build from       |
| <b>Prototype and review</b>       | \$500 – \$2,000    |   | Stakeholder walkthrough, iteration cycles            |
| <b>Developer handoff</b>          | \$500 – \$1,500    |   | Annotated specs, assets, interaction notes           |

### Standard vs custom design

Standard component-based design uses existing platform patterns, Apple's Human Interface Guidelines and Material Design for Android. This is faster, familiar to users, and cheaper to implement. Custom design introduces brand-specific layouts and original visual elements. It creates differentiation but adds cost.

For most business applications, standard design with considered visual branding is the right approach in version one. Reserve custom design investment for specific moments where it genuinely changes the user experience.

## Phase 3: Development

Typical cost: \$20,000 – \$150,000 | 40–55% of total budget | 8–24 weeks

Development is the largest phase by cost and by time. It is also where the most surprises occur when prior phases have not done their job. A well-specified, well-designed project moves through development predictably. An under-specified project generates constant decisions mid-build, each of which costs time.

Development splits into frontend, which users see; and backend, the server, database, business logic, and APIs that power everything. Backend complexity is the primary cost driver, not the number of screens.

### How the development budget is distributed

## Code B | Step-by-Step App Cost Breakdown

| Component                       | Typical share        | What it includes  |
|---------------------------------|----------------------|---|
| <b>Frontend development</b>     | 40–50% of dev budget | Screens, navigation, animations, and device compatibility |
| <b>Backend development</b>      | 35–45% of dev budget | APIs, database, business logic, security                  |
| <b>Third-party integrations</b> | 10–20% of dev budget | Payments, notifications, maps, analytics                  |
| <b>Admin dashboard</b>          | \$5,000 – \$15,000   | Content management, user oversight, reporting             |

## Feature-level cost reference

Because development cost is driven by features, understanding what individual features cost helps build a bottom-up estimate.

| Feature                             | Cost range         | Notes                                      |
|-------------------------------------|--------------------|--|
| <b>User authentication (email)</b>  | \$2,000 – \$5,000  | Login, session management, password reset  |
| <b>Social login (Google, Apple)</b> | \$1,500 – \$4,000  | OAuth flow, token handling                 |
| <b>Payment integration</b>          | \$3,000 – \$8,000  | Stripe/Razorpay, webhooks, refund handling |
| <b>Push notifications</b>           | \$1,000 – \$3,000  | Provider setup, targeting logic            |
| <b>Real-time chat</b>               | \$5,000 – \$15,000 | WebSockets or third-party, message storage |
| <b>Maps and location</b>            | \$3,000 – \$7,000  | Provider integration, real-time tracking   |

|                             |                    |  |
|-----------------------------|--------------------|--|
| <b>Search and filtering</b> | \$2,000 – \$6,000  | Index design, query performance            |
| <b>Admin dashboard</b>      | \$5,000 – \$15,000 | User management, content controls, reports |

## Why backend work costs more than it looks

Every feature that sounds simple on the frontend has a backend component that is more complex than it appears. A payment integration is not done when the checkout screen works it is done when failed cards, expired tokens, duplicate submissions, refund flows, and webhook failures are all handled correctly. A login system is not complete when login works; it is complete when session management, password reset, multi-device handling, and security edge cases are all validated.

Agencies that quote development by counting screens instead of mapping backend requirements consistently underprice the work. When evaluating quotes, ask specifically what is included in the backend scope and whether third-party integrations are fully costed.

## Phase 4: QA and Testing

Typical cost: \$5,000 – \$30,000 | 15–20% of total budget | 2–6 weeks

QA ensures that what was built works correctly, performs reliably, and meets the security standards required. It is also the phase that gets cut most often when projects run over time or over budget. The consequences show up immediately after launch — crash reports, negative reviews, security incidents.

Thorough QA on a mobile app requires testing across multiple device types, OS versions, and network conditions. It means validating every user flow against requirements, testing behaviour under load, and verifying that security controls are effective.

### QA cost breakdown

| Test type                 | Cost range        | Scope  |
|---------------------------|-------------------|--|
| <b>Functional testing</b> | \$2,000 – \$8,000 | Every user flow validated against requirements |

## Code B | Step-by-Step App Cost Breakdown

|                                |                    |   |   |
|--------------------------------|--------------------|---|---|
| <b>Device and OS testing</b>   | \$1,000<br>\$4,000 | – | Multiple screen sizes, iOS and Android versions   |
| <b>Security testing</b>        | \$1,000<br>\$5,000 | – | Auth, data handling, API exposure                 |
| <b>Performance testing</b>     | \$800 – \$3,000    |   | Load behaviour, response times under real traffic |
| <b>User acceptance testing</b> | \$500 – \$2,000    |   | Stakeholder review before release                 |

### The real cost of cutting QA

Emergency fixes in production cost more than the same bugs caught in testing, in direct engineering time and in operational disruption. A critical authentication failure or payment processing error found by real users costs \$5,000–\$15,000 to diagnose, fix, test, and redeploy. The same issue caught in QA is a \$500 task.

Negative App Store reviews citing crashes suppress organic downloads for months after the issue is resolved. User trust lost at launch is difficult to rebuild. QA is not a cost; it is an insurance policy on the entire investment that preceded it.

## Phase 5: Deployment

Typical cost: \$2,000 – \$8,000 | 3–6% of total budget | 1–2 weeks

Deployment is the process of moving the application from development to production and making it available through the App Store and Google Play. It involves more than pressing a submit button. A proper deployment setup includes configuring production infrastructure, setting up monitoring tools, implementing a CI/CD pipeline for future updates, and preparing store assets.

### Deployment cost breakdown

| Item                              | Cost            | Notes   |
|-----------------------------------|-----------------|---|
| <b>App store submission setup</b> | \$800 – \$2,500 | Store assets, metadata, screenshots, descriptions |

|                                  |                    |   |  |
|----------------------------------|--------------------|---|--|
| <b>CI/CD pipeline</b>            | \$1,000<br>\$3,000 | – | Automated build and deployment configuration       |
| <b>Production infrastructure</b> | \$1,500<br>\$4,000 | – | Server config, environment setup, monitoring tools |
| <b>Apple Developer Program</b>   | \$99/year          |   | Mandatory for App Store distribution               |
| <b>Google Play Console</b>       | \$25 one-time      |   | Mandatory for Google Play distribution             |

## App store review timelines

Apple's review process takes 24–72 hours for most apps, with manual human review against their guidelines. Around 40% of submissions require revisions on the first attempt. Google Play uses automated review and typically processes submissions within a few hours.

Apps with payments, account creation, or sensitive data face additional review scrutiny on both platforms. Privacy policy documentation, data handling declarations, and clear functionality descriptions all affect review outcomes. These need to be prepared as part of the deployment phase, not after submission is rejected.

## Phase 6: Post-Launch Maintenance

Typical cost: \$8,000 – \$40,000 in year 1 | 15–25% of original build cost annually | Ongoing

Every app in production requires ongoing maintenance. Operating systems release major updates annually. Third-party libraries and APIs change, deprecate features, or alter pricing. Security vulnerabilities emerge. User feedback reveals issues that testing did not surface. None of these is discretionary.

The standard for annual maintenance budgeting is 15–25% of the original build cost. On a \$60,000 app, that is \$9,000–\$15,000 per year. On a \$150,000 app, it is \$22,500–\$37,500. Where you land depends on how actively the product is being developed, how many third-party integrations require attention, and how quickly the user base is growing.

## Monthly maintenance cost breakdown

| Item                             | Monthly cost      | Notes  |
|----------------------------------|-------------------|--|
| Cloud hosting and infrastructure | \$100 – \$500     | Scales with traffic; Firebase or AWS/GCP       |
| Third-party service fees         | \$50 – \$300      | Payment APIs, SMS, maps, analytics             |
| Bug fixes and stability          | \$500 – \$2,000   | Responsive to live user issues                 |
| OS compatibility updates         | \$200 – \$600     | Annual iOS/Android releases, amortised monthly |
| Security patches                 | \$200 – \$500     | Library updates, vulnerability fixes           |
| Minor feature updates            | \$500 – \$2,000   | Incremental improvements from user feedback    |
| <b>Total typical range</b>       | \$1,550 – \$5,900 | Mid-complexity app, post-launch year 1         |

## What happens without maintenance

Apps that fall too far behind Apple or Android's minimum SDK requirements can be delisted from the App Store. This has happened to real products. Apple issues warnings before enforcement, but teams that ignore them eventually face forced removal, and restoration under time pressure costs more than proactive maintenance ever would have.

Security neglect is the other major risk. Libraries with known vulnerabilities stay in production. Authentication systems are not updated as standards evolve. The cost of a security incident, legal exposure, user notification, and engineering response exceeds years of proactive maintenance.

## Building a Realistic Budget

With all six phases mapped, here is how to construct a project budget that holds through delivery. The most common mistake is budgeting only for development and treating the rest as optional. Every phase in this guide is required for a production-ready product.

### Complete budget framework

| Phase                           | Cost range           | % of budget | Planning note                 |
|---------------------------------|----------------------|-------------|-------------------------------|
| <b>Discovery &amp; Planning</b> | \$3,000 – \$10,000   | 5–8%        | Required — never skip         |
| <b>UI/UX Design</b>             | \$5,000 – \$20,000   | 8–15%       | Scale to complexity           |
| <b>Development</b>              | \$20,000 – \$150,000 | 40–55%      | Largest single line           |
| <b>QA &amp; Testing</b>         | \$5,000 – \$30,000   | 15–20%      | Non-negotiable                |
| <b>Deployment</b>               | \$2,000 – \$8,000    | 3–6%        | Infrastructure + store        |
| <b>Contingency (15%)</b>        | 15% of the total     | 15%         | Scope changes are inevitable  |
| <b>Year 1 maintenance</b>       | \$8,000 – \$40,000   | 15–25%      | Plan before launch, not after |

### The contingency line

A 15% contingency buffer is not pessimism; it is realistic project management. Scope change during development is normal, not exceptional. Technical dependencies turn out to be more complex than estimated. New requirements emerge from stakeholder review. These situations are handled predictably when contingency is built in and poorly when it is not.

### Platform choice and its effect on the budget

Every range in this guide assumes a single-platform build. Building native iOS and Android simultaneously roughly doubles development and QA costs. Cross-platform frameworks like Flutter or React Native share code across platforms and typically reduce total cost by 20–35% compared to two separate native builds. The right choice depends on where your users are.

## Team location and its effect on the budget

The same project built by a US agency (\$150–\$250/hour blended) versus an Indian agency (\$25–\$60/hour blended) produces a cost difference of \$100,000–\$200,000 on a mid-complexity app. That gap is explained by operating costs and salary levels, not by capability differences. The condition for offshore development to deliver on its cost advantage is process discipline: clear specifications, agreed communication rhythms, and milestone-based delivery.

## Five Budgeting Mistakes to Avoid

### 1. Treating discovery as optional

Discovery is 5–8% of the total budget. Skipping it does not save money; it defers the cost of unresolved requirements into development, where addressing them is far more expensive.

### 2. Budgeting only for development

A project budget that includes only development will consistently underrun against reality. Design, QA, deployment, infrastructure, and maintenance are not optional add-ons. They are the complete cost of building and running a production app.

### 3. Accepting a quote without a line-item breakdown

A total project number with no breakdown reveals nothing about what is and is not included. QA, infrastructure, and deployment are frequently excluded without being clearly stated. Ask for a phase-by-phase breakdown before accepting any estimate.

### 4. Building everything in version one

Products that launch with a full feature set before any real user data exists frequently discover that many of those features are unused or need reworking. An MVP that launches the core value proposition at \$30,000–\$50,000 and iterates based on real feedback consistently produces better outcomes than a \$200,000 complete build launched once.

## 5. Not planning maintenance before launch

Maintenance is not a decision to make after launch; it is a budget line to include before launch. Teams that discover post-launch costs after going live either underfund maintenance or cut it entirely, both of which damage the product over time.

---

## About Code B

Code B is a software development agency that builds mobile apps, web platforms, and data engineering solutions for SMEs and product companies across the US, UK, and Europe. Our clients range from founders building their first product to established businesses rebuilding systems that have outgrown their original architecture.

We work across the full development lifecycle, from scope definition and technical architecture through to launch and ongoing maintenance. If you are planning an app and want a clear estimate based on your actual requirements, get in touch at [code-b.dev](https://code-b.dev).

*All cost ranges are indicative and based on standard market rates as of 2026. Actual project costs depend on specific requirements, team structure, and scope.*