Module 3.7: Developer Handoff Guide

Protect your investment with a proper handoff.

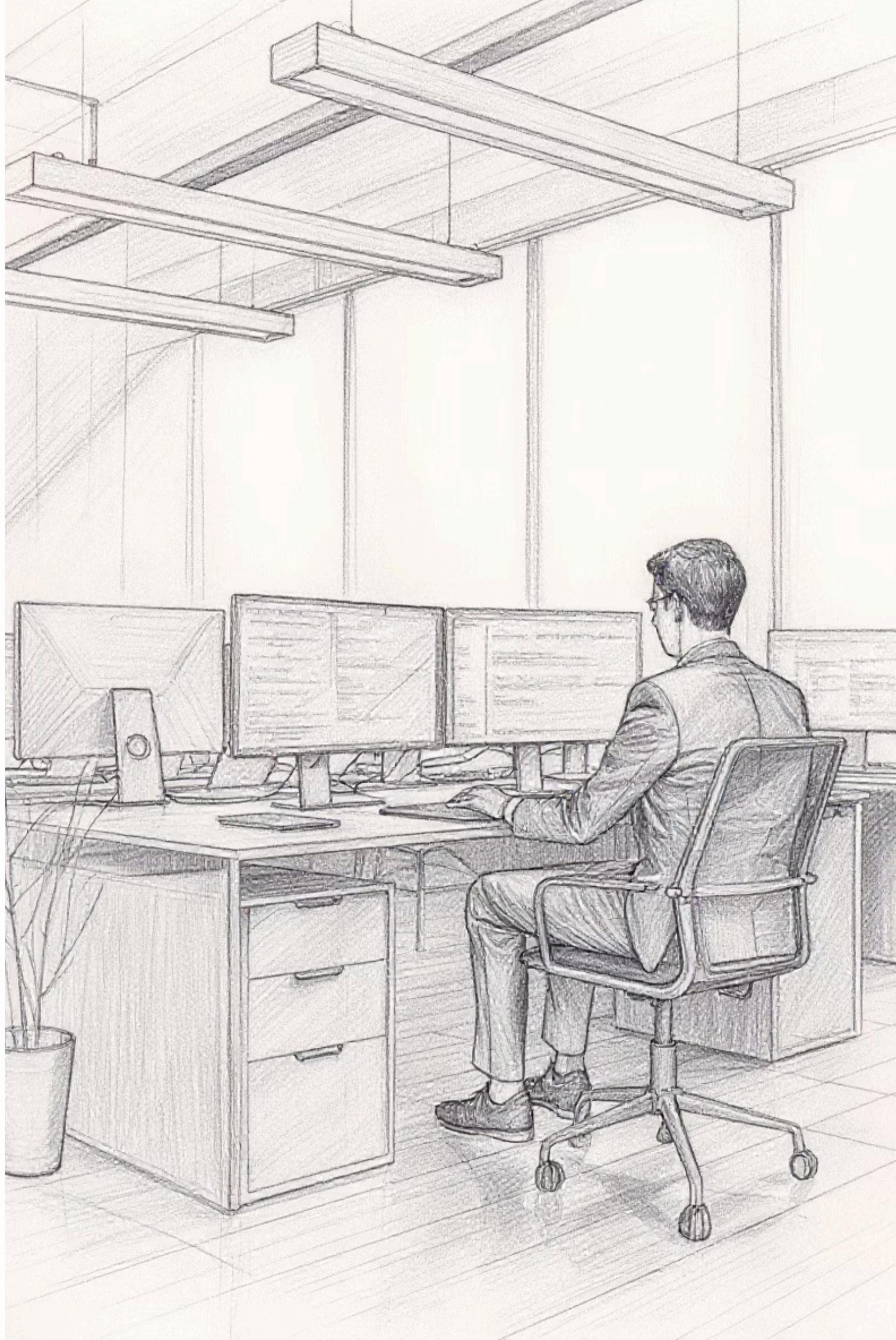
When You Need This

Current Developer Issues

Performance problems or personal reasons

Project Complete

Development finished, now you need ongoing maintenance.



The Cost of Poor Handoffs

😱 Nightmare Scenario

- Developer leaves abruptly
- Code is messy
- New developer can't use it
- Forced to rebuild from scratch

✅ Smooth Scenario

- Handoff materials ready
- New developer onboards in 2 weeks
- Development continues seamlessly
- Zero disruption to progress

Before You Hire: Protection Basics

1

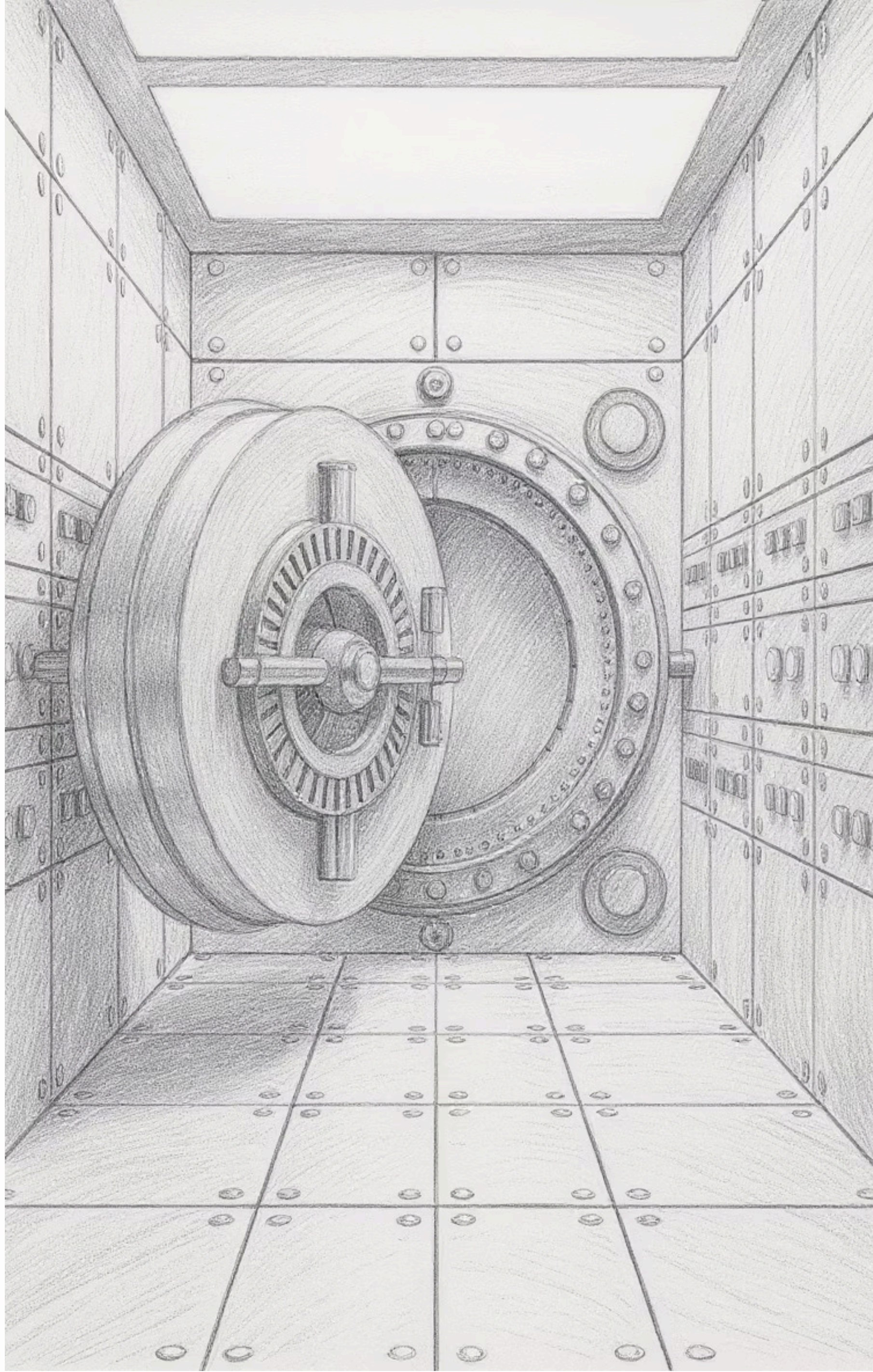
Get Everything prepared:

- Repository access (Github/Gitlab/Bitbucket)
- Admin access to all services
- All API keys and credentials
- Documentation of setup

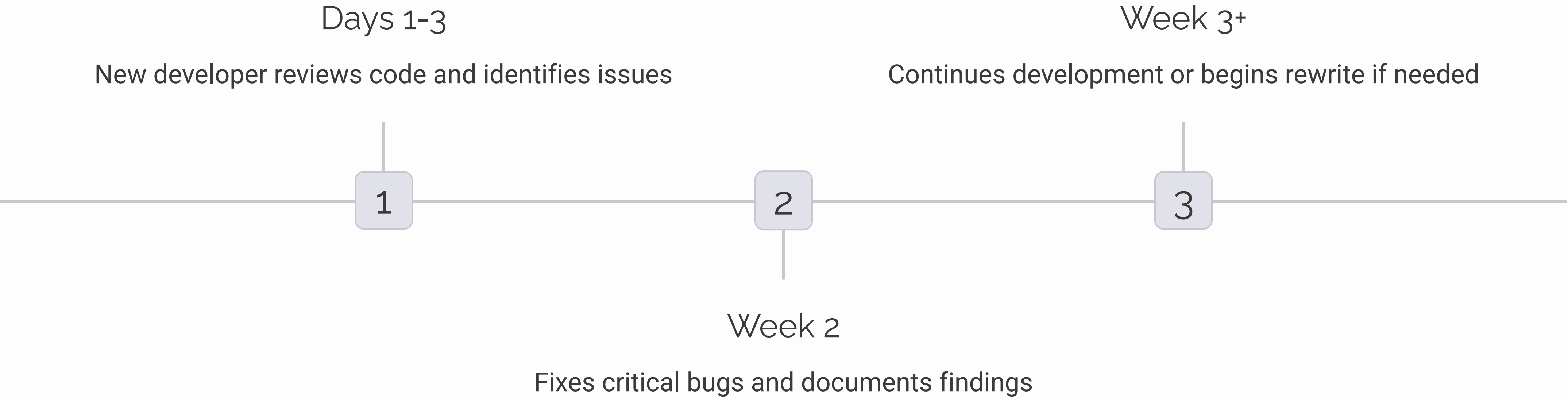
2

Use a Password Manager

Set up 1Password or LastPass for your company.
Add all developer credentials they need.



Transition Timeline & Budget



1

Days 1-3

New developer reviews code and identifies issues

2

Week 2

Fixes critical bugs and documents findings

3

Week 3+

Continues development or begins rewrite if needed

Budget for transition: Expect 20-40 hours of work.

If new developer says "this needs rewriting," get a second opinion from a technical advisor before proceeding.

The Non-Negotiable Handoff Checklist

This should take 2-3 days to compile if the developer has been working properly. If it takes longer, they weren't maintaining documentation.

01

Documentation Package

Complete technical documentation

02

Access & Credentials

All account ownership transfers

03

Working Proof

Verified setup and deployment

04

Knowledge Transfer

Recorded walkthrough sessions

05

Post-Handoff Support

Two weeks of availability

Part 1: Essential Documentation

1

README File

Project overview, setup instructions, deployment process. A different developer should be able to set up the project in under 2 hours using only this document.

2

Environment Variables List

Every API key, token, and secret with instructions on where to get them and what they're used for.

3

Database Schema

All database tables, what each stores, how they relate, and example data.

4

Deployment Process

Step-by-step deployment instructions, hosting details, and rollback procedures.

5

Known Issues List

All known bugs with workarounds, priority levels, and why they haven't been fixed.

6

Feature Roadmap

Planned features with technical complexity, time estimates, and potential challenges.

README Quality Check

❌ Bad README

Clone the repo and run npm install.
Then start the server.

Missing: database setup, environment variables, how to start server, port information, and more.

✅ Good README

```
## Setup Instructions
1. Install Node.js version 18+
2. Clone this repository
3. Copy .env.example to .env
4. Run: npm install
5. Run: npm run setup-db
6. Run: npm start
7. Open browser to localhost:3000
```

Clear, complete, and actionable.

Part 2: Account Ownership

Critical: YOU must own all of these accounts, not the developer.

Repository Access

Transfer GitHub/GitLab ownership to your account. You're the owner, developer is collaborator.

Hosting & Database

Heroku, AWS, DigitalOcean access. Direct database credentials and backup system.

Domain Registration

DNS provider login. Domain registered under YOUR name, not developer's.

Payment Processing

Stripe, PayPal account access, API keys, and webhook configurations.

Email & Analytics

SendGrid, Mailgun, Google Analytics access and API keys.

Third-Party Services

All APIs, authentication services, SMS providers, and image hosting.

Critical Security Step: After handoff, change all passwords, revoke developer's access everywhere, generate new API keys, and update the app.

Part 3: Working Proof Tests

Before accepting the handoff, verify everything works:

Test 1: Local Setup (2 hours max)

New developer sets up project from scratch using only documentation. Must work in under 2 hours without asking questions.

Test 2: Production Deployment (30 min)

Make a tiny change, deploy to production following docs, verify it worked. Should take under 30 minutes.

Test 3: Tests Pass

If automated tests exist, they should all pass. If tests fail, something is broken.

🚨 **If any test fails, documentation is incomplete.** Original developer must fix it before you accept the handoff.

Part 4: Knowledge Transfer Sessions

Record all sessions for future reference.

Session 1: Codebase Walkthrough

2 hours. Original developer walks through architecture, major features, design decisions, and shortcuts taken.

Session 2: Q&A

1-2 hours. After code review, new developer asks questions about approach, pain points, and fragile areas.

Session 3: Edge Cases

1 hour. Cover things that break easily, workarounds, tricky code, and production quirks.

Part 5: Post-Handoff Support

> Build 2 weeks of availability into the contract. Original developer responds to questions within 24 hours via email or Slack.

✅ What This Covers

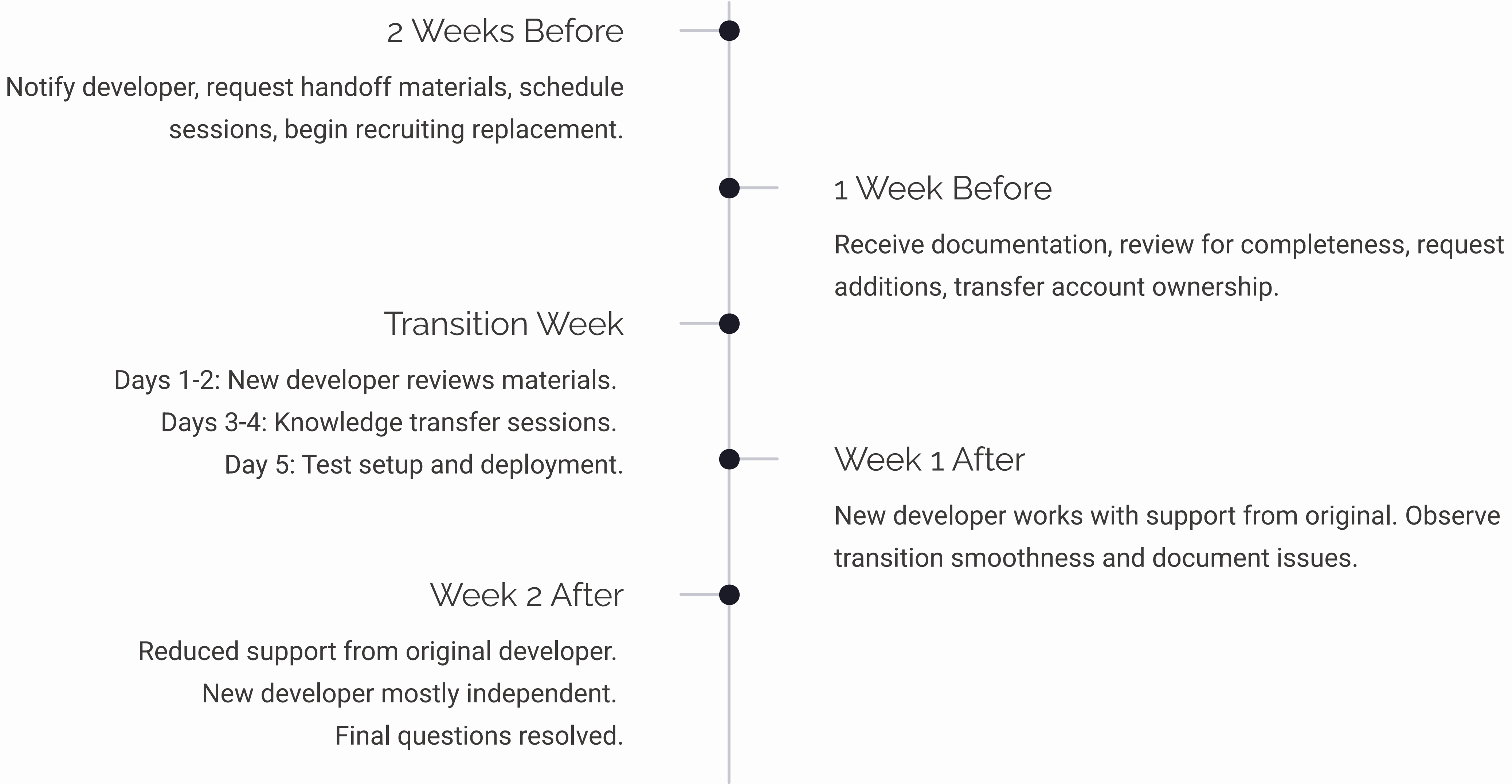
- "How do I do X?"
- "Where is Y located?"
- "Why did you do Z this way?"
- Clarifications on documentation

❌ What This Doesn't Cover

- Building new features
- Fixing old bugs
- Code review of new work

Typical cost: Include in original contract of availability.

The Complete Transition Timeline



2 Weeks Before

Notify developer, request handoff materials, schedule sessions, begin recruiting replacement.

1 Week Before

Receive documentation, review for completeness, request additions, transfer account ownership.

Transition Week

Days 1-2: New developer reviews materials.
Days 3-4: Knowledge transfer sessions.
Day 5: Test setup and deployment.

Week 1 After

New developer works with support from original. Observe transition smoothness and document issues.

Week 2 After

Reduced support from original developer.
New developer mostly independent.
Final questions resolved.

Red Flags During Handoff

🚨 "The code is self-documenting"

Translation: I don't want to write it.

Response: "That's not acceptable. Documentation is part of the deliverable."

🚨 "Any good developer will understand it"

Translation: I didn't organize it well and I'm defensive.

Response: "That's not the point. We need documentation for efficiency. Please provide it."

🚨 "I don't have time to do a handoff"

Translation: I don't want to be accountable for my work.

Response: "This is part of the contract. If you can't provide a proper handoff, we need to discuss payment."

🚨 "Just give them my phone number"

Translation: I didn't document anything and want to charge for future questions.

Response: "No. I need written documentation and formal knowledge transfer sessions."

What to Do If Handoff Is Poor

Option 1: Demand Original Developer Fix It

Withhold final payment, require proper documentation within 1 week. If they refuse, move to Option 2.

Option 2: Budget for Documentation

Pay new developer to create documentation. Typically 15-40 hours.

Treat as sunk cost and learn from it.

Option 3: Rebuild (Last Resort)

Only if code is genuinely unsalvageable AND you have traction/users. Maintain current version while rebuilding. Don't make this decision lightly.

Action Checklist: Before Next Project

Contract Protection

- Add handoff clauses to developer contract
- Require documentation every 2 weeks
- Make final payment contingent on handoff
- Budget for 2 weeks of transition overlap

During Handoff

- Get all materials from checklist
- Transfer all account ownership
- Schedule knowledge transfer sessions
- Record all sessions
- Have new developer test setup and deployment

After Handoff

- Change all passwords
- Revoke developer's access
- Generate new API keys
- Store all documentation safely
- Keep recordings for future reference

The Bottom Line

$10K-30K

Money Saved

A smooth handoff prevents costly rebuilds and wasted development time.

2-4

Weeks Saved

Proper documentation means new developers can start contributing immediately.

100%

Peace of Mind

You own everything and can transition developers without fear.

Key Takeaways

Protect Yourself Upfront

Use password managers and maintain account ownership from day one.

Demand Complete Documentation

Don't accept "self-documenting code." Require written documentation as a deliverable.

Test Before Accepting

New developer must successfully set up and deploy using only the documentation provided.

Record Everything

Knowledge transfer sessions are invaluable for future reference.

Budget for Transition

Plan for 20-40 hours of transition work and 2 weeks of overlap support.

Secure After Handoff

Change all passwords and revoke access immediately after transition.

Remember: A smooth handoff saves you $10K-30K and weeks of time.

Demand it. Your business depends on it.

Download Checklist

Share This Guide