

Module 3.2: Contracts

What You Need Before Starting Development

Contracts protect your intellectual property, define clear expectations, and provide security if things go wrong.

Three critical agreements every startup needs:

- NDAs to protect your ideas
- consultant agreement
- development contract



Non-Disclosure Agreement (NDA)

An NDA protects your sensitive information when sharing it with developers or agencies.

It's your first line of defense for trade secrets, source code, and business plans.

01

Parties & Purpose

List both sides clearly:

- you as the Disclosing Party
- developer/agency as the Receiving Party

02

Define Confidential Information

Cover all sensitive materials:

- source code,
- business plans, financials
- technical specs, and designs.

03

Receiving Party Obligations

They must keep all info strictly confidential, use it only for project evaluation or execution.

04

Exclusions & Term

Info isn't confidential if publicly available, already known, or obtained legally elsewhere.

05

Return & Jurisdiction

They must return or delete all confidential info upon request. Define applicable law and court jurisdiction.

Development Contracts

Tech Consultant Agreement

- Identify parties and purpose clearly
- Define exact services and deliverables in Scope of Work
- Specify payment terms: hourly, daily, or fixed rate
- Set duration and termination rules (e.g., 30 days' notice)
- Include confidentiality and NDA provisions
- Ensure all deliverables belong to you once paid
- State consultant is independent contractor, not employee
- Limit liability and define governing law

Development Agency Contract

- Define project model: Fixed Price, Time & Material, or Dedicated Team
- Detail scope of work, features, and milestones
- Set timeline, deadlines, and payment triggers
- Outline change request approval process
- Ensure you own all code, designs, and assets
- Define bug-fixing period (30-60 days post-delivery)
- Include acceptance criteria and code quality standards
- Specify communication frequency and reporting

📄 **Pro Tip:** for agency work, often use two documents:

- Master Service Agreement covering how you'll work together (IP, non-solicitation, notices)
- Scope of Work Agreement detailing exact technical specs, delivery plan, price, deadline, and quality KPIs.

Critical Clauses That Protect You

These essential contract provisions safeguard your interests and prevent common pitfalls that can derail your project.



IP Ownership

"All code, designs, and materials created during this project are work-for-hire and owned by [Your Company]. Developer assigns all rights immediately upon creation."

Why: Without this, the developer legally owns the code they wrote for you.



Code Escrow & Repository Access

"Developer will commit all code to a repository controlled by [Your Company] at least weekly. [Your Company] maintains admin access at all times."

Why: If the developer disappears, you need immediate access to your code.



Documentation Requirements

"Developer will maintain README documentation including setup instructions, deployment process, third-party services list, and API keys. Updated weekly."



Trial Period with Early Exit

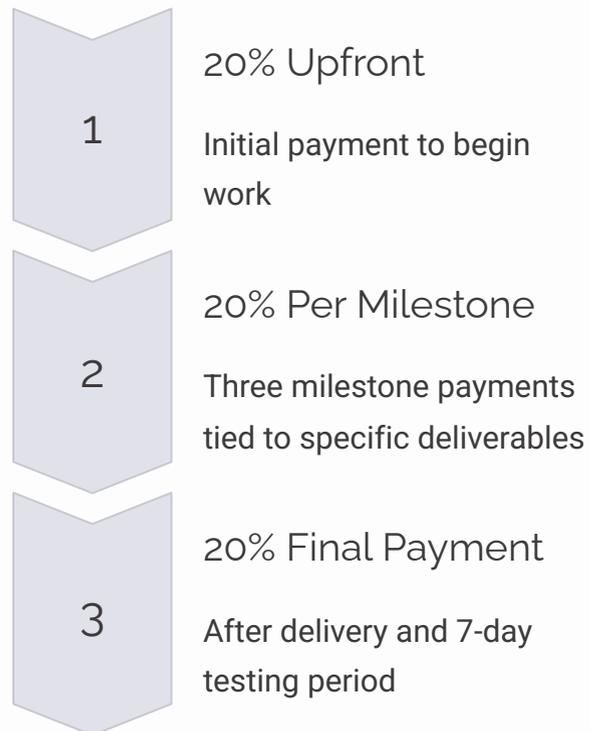
"Either party may terminate within first 30 days with 7 days notice and payment only for work completed."

Payment Structure & Completion Standards

Payment Structure

Never pay 50% upfront.

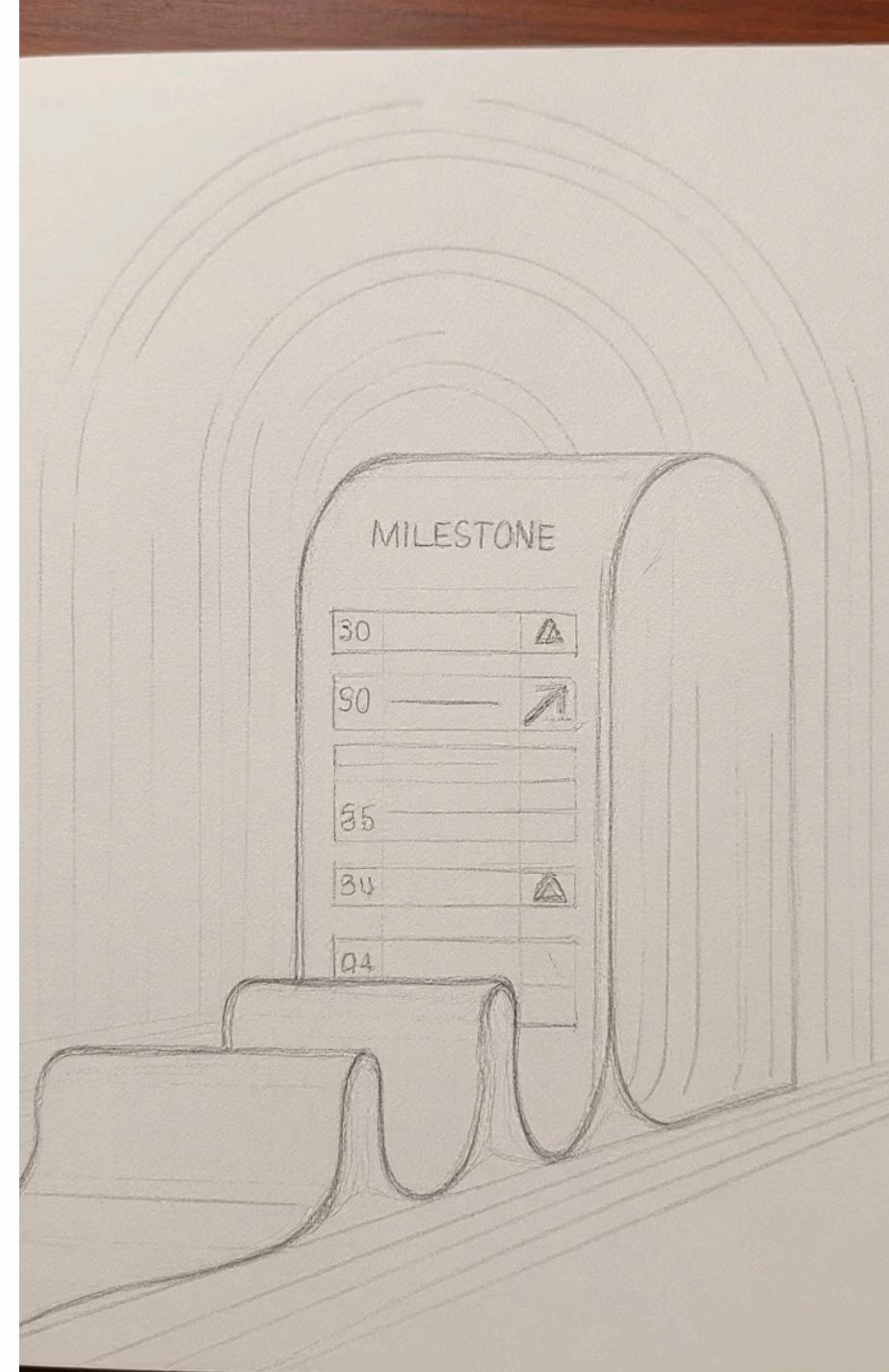
Instead, structure payments around milestones:



Definition of "Complete"

Define it explicitly in your contract:

- Works on specified devices and browsers
- Your company has tested it thoroughly
- All bugs found in testing are fixed
- Code is committed to repository
- Documentation is updated



Post-Launch Support: Your Options After MVP

1

Retainer Agreement

- **Costs:** High, fixed monthly fee, potentially covering a set number of hours or specific services.
- **Benefits:** Predictable budget, priority support, quick response times, ongoing feature development or improvements.
- **Watch for:** Ensure the SoW is clearly defined. Address how unused hours (if applicable) are handled.

2

Bug-Fix Only

- **Costs:** Lower, often billed hourly or per incident, only when issues arise.
- **Benefits:** Cost-effective for stable products, pays only for critical problem resolution, dedicated focus on stability.
- **Watch for:** Response times for non-critical issues might be slower. Lacks proactive maintenance or updates.

3

Maintenance

- **Costs:** Moderate, often a fixed monthly (uptime monitoring, security patches, minor updates).
- **Benefits:** Access to expertise and scalable resources. More cost-effective than a full in-house team for ongoing routine tasks.
- **Watch for:** Potential lack of deep context. Ensure Service Level Agreements (SLAs) are robust and clear.

4

In-House Transition

- **Costs:** High initial setup, ongoing salaries, benefits, and infrastructure.
- **Benefits:** Full control over development, product knowledge within the company, immediate response capability.
- **Watch for:** Significant recruitment effort and cost. Requires substantial internal expertise and resources.

What You Actually Need: First 6 Months Post-Launch

For the initial six months after your MVP launch, focus should be on stability and minor refinements, not heavy feature development. Your post-launch support should primarily cover:

- Bug fixes (should be rare if built well)
- Minor tweaks (e.g., button colors, text changes)
- Analytics and tracking adjustments

This limited scope means **post-launch support should not exceed \$500-\$1,000 per month.**

If your post-launch costs are consistently higher, it indicates one of three core issues:

Unstable MVP

Product may have been released with underlying issues.

Overcharging

Developer might be billing excessively for routine tasks or inflating hours.

Feature Creep

You are adding new features or scope under the "support."