

# The Complete Outsourcing Due Diligence Checklist

Evaluating, Hiring, and Managing an Outsourced Software Team

**Dr James Cannan**

Technical Founder · Startup Coach · Fractional CTO

[jamescannan.com](http://jamescannan.com)

## How to Use This Guide

---

### The Problem With Outsourcing Software

Outsourcing software development is one of the highest-risk decisions a founder or business owner will make. Done well, it can accelerate your product and give you access to world-class talent at a fraction of the cost of building an in-house team. Done badly, it can drain your budget, destroy your timeline, and leave you with code you cannot maintain or even own.

The reality is that the outsourcing market is full of agencies that are excellent at selling and poor at delivering. They show you polished case studies, speak fluent startup language, and quote you a price that feels almost too good. Then work begins, communication slows, quality slips, and you find yourself six months in with half a product and no clear path forward.

These are not edge cases. They are the most common outcomes for founders who approach outsourcing without a structured evaluation process.

**The best time to vet a team is before you pay them anything.**

### Where It Goes Wrong

Most outsourcing failures share the same root causes. Founders choose on price rather than fit. They skip due diligence because the sales process felt reassuring. They sign contracts without protecting their IP. They start work without agreeing what 'done' means. And they discover the problems only after the money has been spent.

The specific warning signs are predictable once you know what to look for: a team that cannot produce verifiable client references, developers you never speak to before signing, fixed-price quotes delivered without discovery, vague answers about testing and process, and scope that creeps without any formal change control.

Understanding these patterns is half the battle. The other half is having a process rigorous enough to surface them before they cost you.

### Why a Trusted Partner Makes All the Difference

The founders who have the best outsourcing experiences are rarely the ones who found the cheapest team. They are the ones who found a team they could genuinely trust, a team that communicated clearly, delivered consistently, and behaved like a partner rather than a vendor.

When you get this right, outsourcing becomes a genuine competitive advantage. You can move faster than competitors building in-house. You can access specialist expertise on demand. You can scale development up and down without the overhead of employment. The checklist in this guide is designed to help you find that kind of team and avoid the ones that will waste your time and money.

### How to Use This Checklist

Work through each of the seven sections before committing to an outsourced team. Use it during your evaluation process, not after you have already decided.

Items marked [CRITICAL] are non-negotiable. If a team fails these checks, walk away regardless of how good the pitch was.

Items marked [WATCH] are yellow flags. They do not automatically disqualify a team, but they require a clear and satisfactory explanation before you proceed.

# The Checklist

## Section 01

### Vetting the Team

---

- Verify the company has been operating for at least 3 years  
*Longevity reduces the risk of a team dissolving mid-project.*
- Review projects similar in scope and technology to yours **[CRITICAL]**  
*Ask for live URLs, or case studies, not just screenshots.*
- Speak with past clients (not references they hand-pick)  
*Ask about timelines, communication quality, and how problems were handled.*
- Check independent reviews on Clutch, Trustpilot, or Google  
*Look for patterns in criticism, one bad review is noise, three is a signal.*
- Assess English proficiency across the whole team, not just the sales contact  
*Request a video call with the developers or tech lead you will actually work with.*
- Understand employee vs contractor mix, will your work be subcontracted? **[WATCH]**  
*Some agencies subcontract without disclosure, adding quality and IP risks.*

## Section 02

### Technical Capability

---

- Request a technical discovery call with the lead developer, not just a PM **[CRITICAL]**  
*Assess depth of knowledge, ask how they would architect your specific project.*
- Confirm hands-on experience with your specific tech stack  
*Enthusiasm for a new technology is not the same as production experience.*
- Ask how they handle automated testing, what is their code coverage policy? **[WATCH]**  
*Teams without a testing culture create expensive technical debt.*
- Ask about their version control workflow and branching strategy  
*Git Flow, trunk-based, feature branches, any structured answer is positive.*
- Request to see a sample of code from a past project (redacted if needed) **[WATCH]**  
*Code quality, naming conventions, and comments reveal engineering maturity.*
- Confirm experience with your deployment environment (AWS, Azure, GCP, etc.)
- Ask how they handle security, OWASP, dependency audits, secrets management **[CRITICAL]**  
*Security is often an afterthought for less experienced teams.*
- Understand their approach to documentation and knowledge handoff  
*You need to be able to maintain or hand over the code after the project ends.*

## Section 03

## Process & Project Management

---

- Confirm what methodology they use (Agile, Scrum, Kanban) and get specifics  
*Vague answers like 'we are flexible' often mean no process at all.*
- Agree on sprint length, ceremony cadence, and how progress is tracked **[CRITICAL]**  
*Weekly demos are the minimum, you should see working software regularly.*
- Establish who your single point of contact is and their typical response time **[WATCH]**  
*Lack of a clear PM is a major early warning sign.*
- Understand their escalation path, what happens when something goes wrong?
- Agree on the project management tool and confirm you will have full access  
*Jira, Linear, Asana, you should see the actual backlog, not just status updates.*
- Define how requirements changes are handled (change requests, impact on cost) **[CRITICAL]**  
*Scope creep handled poorly is the number one reason outsourced projects go over budget.*
- Ask how they handle bugs found after a feature is marked complete  
*Reputable teams include a bug-fixing period as part of each delivery, not an extra cost.*
- Confirm their testing and QA process before any code is delivered to you

## Section 04

## Contracts & Legal Protection

---

- Ensure intellectual property is explicitly assigned to you upon payment **[CRITICAL]**  
*Default IP ownership laws vary by country, do not assume you own what they build.*
- Include an NDA covering all aspects of your product and business
- Include a warranty or defect liability period (minimum 30–90 days post-launch)  
*Covers bugs discovered after delivery that were not caught during testing.*
- Specify termination conditions and what happens to code and data if you part ways **[CRITICAL]**  
*You must be able to obtain all source code and data on termination, even mid-project.*
- Ensure they carry appropriate professional indemnity insurance
- Define data processing responsibilities if handling personal data (GDPR, CCPA) **[WATCH]**  
*The outsourced team may need to be a registered data processor.*

## Section 05

## Communication & Collaboration

---

- Confirm overlapping working hours and the minimum daily overlap window **[WATCH]**  
*Fewer than 2 hours of real-time overlap per day creates significant delays.*
- Agree on a communication platform and expected response times  
*Slack, Teams, or email, define it clearly from day one.*
- Schedule a recurring weekly video call, not just async updates  
*Video communication catches misalignment far earlier than text threads.*
- Define what a status update must contain (blockers, completed, next steps) **[WATCH]**  
*Vague 'everything is going well' updates are a red flag.*
- Establish how design feedback and wireframe reviews will be handled
- Agree on how urgent issues are escalated outside normal hours

## Section 06

## Before Work Begins

---

- Conduct a paid discovery or scoping phase before committing to the full build **[CRITICAL]**  
*A small paid engagement reveals how they work far better than any sales call.*
- Agree on a Definition of Done for every deliverable before development starts  
*Ambiguity about 'done' causes the most heated disputes.*
- Set up all shared tools and access before day one (repo, PM tool, comms)  
*Delays in access setup in week one signal poor onboarding maturity.*
- Establish a clear handoff protocol for credentials, environments, and secrets  
*Never share passwords over email or Slack messages.*
- Define who owns final sign-off at each project milestone on your side  
*Slow internal approvals on your end are just as damaging as delays on theirs.*
- Create a simple onboarding document covering your product, users, and goals  
*Do not assume they will ask the right questions — give them the context up front.*

## Section 07

## Red Flags — Walk Away If You See These

---

- They cannot provide verifiable client references or past client contact details **[CRITICAL]**
- The quote comes back unrealistically fast with a fixed price and no discovery **[CRITICAL]**  
*Complex software cannot be accurately estimated without detailed requirements.*
- They refuse to sign an NDA before discussing your project in detail **[CRITICAL]**
- They propose paying for the entire project upfront **[CRITICAL]**  
*Legitimate agencies use milestone or time-based billing — not full upfront payment.*
- They pressure you to sign quickly with time-limited pricing **[WATCH]**  
*A good agency is not short on work. Urgency tactics are a sales strategy.*
- Key people change between the sales process and project kickoff **[WATCH]**  
*The team you evaluated is the team you expect to build your product.*
- They cannot explain their testing process in concrete terms **[WATCH]**
- Communication drops off or becomes vague during the proposal phase **[CRITICAL]**  
*How they communicate before you pay predicts how they communicate after.*

### Need Help Navigating This?

I have done this before, and have a team I use in India. If you would like a free introduction. Please let me know.

[jamescannan.com](https://jamescannan.com)