DIVSPHERE · FIELD GUIDE

Scoping a Fixed-Price Software Project

A practical checklist for founders and operators buying custom software — what a real scope document needs, the questions to ask before you accept a quote, and the red flags that predict a change-order fight later.

Why this guide exists

Most fixed-price software engagements that go wrong don't go wrong because of bad engineering. They go wrong because of a scope that was agreed to before anyone had really defined it. A vendor quotes a number against a one-paragraph description, both sides nod, and three months later there's a change-order fight over whether "user management" was supposed to include password reset, role permissions, and audit logging — or just a login screen.

This guide is the checklist we walk clients through before we quote a fixed-price engagement. It exists so you can run the same process whether you're evaluating us or another vendor — the goal is a scope document precise enough that a fixed price actually means something.

Part 1: What "scope" actually needs to contain

A scope document that's just a features list is not a scope document — it's a wish list. A specification a vendor can accurately price against needs five things:

1. User roles and what each one can do. Not "admins and users" — the actual roles in your business (store manager, warehouse staff, regional supervisor, customer) and a table of what each can see, create, edit, and delete. This single artifact eliminates more scope disputes than anything else on this list.

2. Every screen, named. A list of every screen or view the system needs, even placeholder names ("Order detail page," "Admin — user list," "Password reset flow"). If a screen isn't on this list, it isn't in scope. This is also where you catch the screens nobody thought about — the empty state, the error state, the "no results found" state.

3. Integrations, with their failure modes. Every third-party system the software talks to — a payment gateway, an SMS provider, an existing ERP, a shipping API — needs to be named, along with what happens when that integration is slow or down. "What happens if the payment gateway times out" is a five-minute conversation before the contract is signed and a two-week argument after.

4. Data migration, if any exists. If you have existing data — a spreadsheet of customers, a legacy database, a competitor's export — say so explicitly, with an estimate of row counts and data quality. Vendors routinely underquote migration because "just import the CSV" turns out to mean "the CSV has four different date formats and four thousand duplicate rows."

5. What "done" looks like, in writing. Acceptance criteria — the specific, testable conditions under which each feature is considered complete. Not "the checkout flow works" but "a customer can add items to a cart, apply a discount code, pay via the two configured methods, and receive a confirmation email within 60 seconds."

Part 2: The questions to ask before you accept a quote

If a vendor hands you a fixed price without asking most of these, the price is a guess, not an estimate — and guesses get renegotiated mid-project.

- How many user roles does the system need, and are permissions per-role or configurable per-user?
- Which third-party integrations are required, and has the vendor worked with those specific APIs before?
- Is there existing data to migrate, and has anyone actually looked at its quality?
- What's explicitly *out* of scope? (A good vendor will proactively list this — it's the fastest way to prevent disputes.)
- How are change requests during the project priced and approved?
- What does the payment schedule look like, and is it tied to milestones or to the calendar?
- Who owns the source code, and when does that ownership transfer?
- What's the warranty or bug-fix period after launch, and what counts as a "bug" versus a "new feature"?
- Will you get staging/UAT access during development, or only a single delivery at the end?

Part 3: Red flags in a scoping conversation

The vendor doesn't ask about your users. A vendor pricing a project without asking who uses each part of the system, and how often, is pricing based on assumptions rather than your actual business.

The estimate arrives suspiciously fast. A detailed fixed-price quote for a genuinely custom system, delivered within a day of a single call, usually means either the vendor is padding heavily to cover the unknowns, or they haven't actually thought it through and the number will move later.

Nothing is written as "out of scope." If every conversation ends with "yeah, we can include that," but nothing is ever excluded in writing, the eventual invoice for "extras" will be larger than expected.

No mention of a change-request process. Requirements evolve on every real project — that's normal. What matters is whether there's an agreed process for pricing and approving changes before they happen, versus discovering the process exists only after the first dispute.

The contract has no acceptance criteria. If "done" isn't defined per feature, the fixed price is fixed only on the vendor's side. You have no lever to say a delivered feature doesn't count.

Part 4: A minimal scope template

Use this structure as a starting point for your own spec, or hand it to a vendor and ask them to fill in gaps before quoting.

```
PROJECT: [name]

1. USER ROLES
   Role name | Can view | Can create | Can edit | Can delete

2. SCREENS / VIEWS
   [ ] Screen name -- one-line purpose -- which role(s) can access it

3. INTEGRATIONS
   Service | Purpose | Failure behavior required

4. DATA MIGRATION
   Source | Approx. row count | Known quality issues | Owner of cleanup

5. OUT OF SCOPE (explicit)
   - ...

6. ACCEPTANCE CRITERIA (per feature)
   Feature | Testable condition for "done"

7. PAYMENT SCHEDULE
   Milestone | % of total | Trigger condition

8. POST-LAUNCH
   Warranty period | Bug-fix SLA | Change-request pricing model
```

Closing note

None of this is about distrust of vendors — it's about making the fixed price mean the same thing to both sides of the table. A scope this specific takes longer to write than "build me a CRM," but it's the difference between a fixed-price project that stays fixed-price, and one that turns into a change-order negotiation every few weeks.

If you're scoping a project right now and want a second pair of eyes on the spec before you send it out for quotes, we're happy to look at it — no charge, no obligation. Reach us through the contact page at divsphere.co/contact.