

# Scope, Contracts & Project Agreements — Workbook

This workbook turns the course into a working set of documents you can send to a real client. Complete each section as you finish the matching module, and by the end you will have a deliverables matrix with acceptance criteria, a reusable statement of work, a service agreement with the clauses that decide disputes, and a change-order process that converts scope creep into billable revenue. Use the templates to scope your next project, build your contract, and log every change.

## Scope: Defining the Work Before You Commit

Convert a fuzzy brief into a counted deliverables matrix with acceptance criteria, explicit exclusions, and MoSCoW priorities.

### Worksheet: Deliverables Matrix Builder

For your next project, list every deliverable as its own row and define done for each. Use the SOW Builder template to fill the matrix. If you cannot count it or test whether it is finished, it is not scoped yet.

Deliverable name (plain, specific)

---

Quantity (counted, not a range)

---

Format and file types delivered

---

Acceptance criteria (2-3 testable conditions for done)

---

Client dependency (what must be supplied before this can start)

---

Milestone this deliverable belongs to

---

### Exercise: Find the Ambiguity in Your Last Brief

Take a recent or current project brief and hunt for every place a difficult-but-honest client could interpret it differently than you intend. This is the single pass that removes most future conflict.

- Which deliverable is named so vaguely that the client and you might picture different things (like the logo example)?

---

- Where did you write soon, a few, reasonable, or as needed instead of a date, count, or named limit?

---

- What adjacent work might the client assume is included that you never intended to do?

---

- Which inputs are you silently assuming the client will provide, and what happens if they do not?

---

## Worksheet: Exclusions, Assumptions, and MoSCoW Sheet

Make the boundary of the project explicit and rank every deliverable so trade-offs are easy under pressure. A client who marks everything a Must have has not prioritised; resolve that before you sign.

Exclusions (printing, hosting, copy, stock licenses, maintenance, extra formats)

---

Assumptions (inputs, rights, decision-maker availability, fixed brief)

---

Must have deliverables (project fails without these)

---

Should have deliverables (important, painful to omit)

---

Could have deliverables (first to drop under pressure)

---

Will not have this time (parked for a future phase)

---

## Checklist: Scope Readiness Checklist

- Every deliverable is named, counted, and has a format
- Each deliverable has 2-3 testable acceptance criteria that define done
- An exclusions list names the adjacent work that is not included
- An assumptions list states the inputs and rights I am relying on
- Deliverables are ranked with MoSCoW so trade-offs are pre-agreed
- The scope passes the stranger test: someone new could build exactly this

## Statements of Work and the Agreement Stack

Assemble the eight-section statement of work, wire it to a master agreement, and strip out the ambiguity traps.

## Worksheet: Statement of Work Assembler

Build your reusable SOW by completing all eight standard sections. Keep a master template with bracketed placeholders so you can produce a tailored SOW for a new project in under an hour.

Background and objectives (the client goal in 2-3 sentences)

---

Scope and deliverables (reference the deliverables matrix)

---

Out of scope (exclusions, assumptions, dependencies)

---

Timeline and milestones (named dates for start, reviews, final delivery)

---

Roles and responsibilities (who does what on both sides)

---

Fees and payment schedule (total, deposit, milestone amounts, due dates)

---

Change control reference (one line pointing to the change-order process)

---

---

Acceptance and sign-off (how deliverables are reviewed and who signs)

---

### Exercise: Decide Your Document Stack

Choose how you will structure your paperwork for repeat versus one-off clients, and wire the documents so none contradicts the others.

- For a repeat client, will you sign one master service agreement and attach short SOWs per project?
  - For a one-off client, which clauses are legal (belong in the MSA) and which are operational (belong in the SOW)?
  - How will your SOW reference the governing agreement, and which document controls if they conflict?
  - Which content currently lives in the wrong document and should move (legal terms out of the proposal, deliverables out of the contract)?
- 

### Checklist: Ambiguity-Trap Hunt Checklist

- No etc or similar appears anywhere in the deliverables list
- Every as needed and as required is replaced with a quantity or trigger
- Reasonable revisions is replaced with a counted number of rounds and an extra-round rate
- Including but not limited to does not appear inside any list of what I will deliver
- Every soon, ASAP, and shortly is replaced with an actual date
- Ongoing or as long as needed is bound to a period or a number of hours

## The Service Agreement: Clauses That Protect Both Parties

Set money terms that protect cashflow, decide ownership and licensing, and cap risk with the legal clauses that decide disputes.

### Worksheet: Payment Terms Designer

Lock the money structure before your next invoice goes out. Use the Payment Schedule sheet in the SOW Builder template to set the numbers, and tie ownership transfer to full payment.  
Deposit amount and timing (50% up front or thirds on milestones)

---

Milestone billing points and amounts

---

Net terms (a short specific period such as due within 14 days)

---

Late fee (interest such as 1.5% per month or a flat charge)

---

Work-stop trigger (the point at which unpaid invoices pause all work)

---

What is withheld until final payment (final files and rights transfer)

---

Accepted payment methods and who covers processing fees

---

---

## Exercise: Set Your IP and Licensing Position

Decide deliberately whether each engagement is a full transfer or a license, and price the difference. A full buyout is worth more than a limited license and should cost more.

- Is this project a full copyright transfer on payment, or a license that lets you reuse or relicense?  
\_\_\_\_\_
- If a license, is it exclusive or non-exclusive, and what media, territory, or duration does it cover?  
\_\_\_\_\_
- Will you reserve portfolio rights, and what would you charge a client to keep the work confidential?  
\_\_\_\_\_
- How will you state that third-party stock, fonts, and plugins pass to the client subject to their own terms, not owned outright?  
\_\_\_\_\_

## Checklist: Protective-Clause Checklist

- Ownership of deliverables transfers only on full payment
- Limitation of liability caps my exposure at fees paid and excludes consequential damages
- Indemnity is mutual: the client warrants the rights to assets they supply
- My warranty is narrow: original work and professional delivery, no guarantee of business results
- A termination clause sets notice and what is owed on exit
- A kill fee covers committed time if the client cancels partway through
- Payment, confidentiality, IP, and liability clauses survive termination

## Change Control, Signing, and Enforcement

Stand up the change-order process, adopt an e-signing and storage workflow, and prepare a calm escalation ladder for breach.

### Worksheet: Change-Order Workflow Plan

Design the five-step process you will run on every new request so scope creep becomes approved billable work. Use the Change-Order Log template to record each one. The rule is absolute: no change order, no change.

How you capture the request in writing and confirm it back to the client

\_\_\_\_\_

How you assess impact (additional time, cost, and timeline shift)

\_\_\_\_\_

What your change-order form records (change, new fee, revised dates)

\_\_\_\_\_

How the client approves before new work starts (e-sign or written confirmation)

\_\_\_\_\_

How you update the SOW, schedule, and invoice afterward

\_\_\_\_\_

Your warm script for raising a change order without friction

\_\_\_\_\_

### Exercise: Choose Your Tools and Storage System

Select the tools that will close and store your agreements, and design a storage habit you can rely on under pressure. The moment you need a signed contract is during a dispute, so it must be retrievable in a minute.

- Will you use a pure e-signature tool (DocuSign, Dropbox Sign, PandaDoc) or an all-in-one (Bonsai, HoneyBook)?  
\_\_\_\_\_
- Which vetted template will you start from (AIGA Standard Form, Contract Killer, Docracy)?  
\_\_\_\_\_

- What is your folder and filename convention so the signed date and client are visible at a glance?
- How will you ensure you save the executed copy itself, not just a live link that could change?

---

### Worksheet: Escalation Ladder Script

Write out your gradual, documented response to a breach so you are reading from a plan rather than improvising under stress. Keep every step in writing, calm, and factual.

Friendly reminder wording (day one past due, assumes oversight)

---

Firm notice wording (cites the clause, the late fee, the work-stop trigger)

---

Formal demand wording (references the agreement, the amount, a deadline)

---

When you invoke the work-stop clause and withhold files and rights

---

External options you would consider (small claims, mediation, collections) and the threshold amount

---

### Checklist: Dispute-Prevention Checklist

- I always take a deposit, so I am never working entirely at risk
- I bill milestones, so leverage stays balanced throughout the project
- I release final files and rights only on full payment
- Every deliverable has acceptance criteria, so done is provable not debatable
- A change-order process is in place and I never do unlogged out-of-scope work
- Every agreement is e-signed and stored where I can find it in a minute

### Your Action Plan

1. Build a deliverables matrix for your next project with acceptance criteria for every item
2. Write an exclusions and assumptions list and rank deliverables with MoSCoW
3. Assemble a reusable eight-section statement of work template with bracketed placeholders
4. Run the ambiguity-trap hunt on your draft and replace every vague word with a number, date, or limit
5. Set payment terms with a deposit, milestones, net period, late fee, and work-stop clause
6. Decide your IP position per project and tie ownership transfer to full payment
7. Add limitation of liability, mutual indemnity, narrow warranties, and a kill fee to your contract
8. Adopt a one-page change-order form and commit to no change order, no change
9. Choose an e-signature tool and a consistent storage convention for every signed document
10. Write your escalation ladder scripts so a breach meets a plan, not improvisation









