

Before they trust your code, they read your messages.

When a client messages you on Upwork — the first hello or the hundredth update — you're not just the developer, you're the whole company in a text box. This is how to handle every message so the work is yours to win and yours to keep.

— START HERE

Right now, the message *is* the product.

You're brilliant at building — but the client can't see any of that yet. All they have is how you reply. A fast, clear, honest message is what turns a lead into a project and a project into a repeat client. So treat the conversation with the same standard as the code: it's part of the work, not a distraction from it.

You don't need to sound like a salesperson or write a lot. You need to sound like someone who is **quick, straight, and easy to work with**. That's what wins.

And beneath every habit in this doc sits one thing that has to be genuine: **a sincere desire to help**. We work to aid and assist our clients — to leave them better off, not just to close a task. When that care is real, the fast replies and honest answers follow on their own.

— HEADS UP

The bids are going out — clients will message you.

Our team is bidding for you on Upwork right now. That means real clients will start messaging *you* directly, often within minutes of a proposal landing. Every notification is a live person deciding whether to hire you.

Reply within five minutes — every time, including after office hours. Clients post from every time zone, so their working day is often your evening. A reply within 5 minutes makes us about **9 times more likely to win the client** than a reply at 10 minutes. On Upwork the first freelancer to respond usually gets the conversation, and a fast, human reply beats a slow, perfect one every time. You don't need the full answer yet; you just need to show up.



Install the Upwork mobile app — today

Turn on push notifications so a message reaches you the second it arrives, not the next time you sit at your laptop. It's the single easiest way to hit the five-minute mark while you're heads-down coding.



Your first 3 months: watch Upwork notifications closely

For at least your first three months after joining, keep a careful eye on your Upwork notifications — in office hours and after them. This is when your profile builds its first reviews and reply-time record, and every missed or slow message costs a client you won't get back.

— THE GAME PLAN

Small projects now, the big ones next.

High-quality clients rarely hand a big, long-term project to a freelancer with an empty profile — they need proof first. So the opening move isn't to chase the largest job; it's to earn trust. Bid on a few small projects, deliver them brilliantly, and collect the reviews that make bigger clients comfortable.

Bid small

STEP ONE

Take on a few small, quick jobs. Early on, chase the review — not the dollar amount.

2–3 reviews

THE REAL GOAL

A handful of genuine five-star reviews is what turns an empty profile into a trusted one.

Big projects

THE PAYOFF

With proof on your profile, the larger, long-term engagements start coming to you.

At the start, the dollar amount barely matters — **the review is the prize**. Once two or three real five-star reviews are on your profile, the big projects tend to follow quickly.



Don't wait to be awarded — build the prototype

For a small test project (say one to three days of work), start building straight away instead of waiting for the client to pick you. On these auditions the first freelancer to show a working prototype usually wins the award — a live demo beats ten promises. This is the one exception to "contract first": a quick prototype is your pitch, but the real, billable build still waits for the signed contract.



Only where you're sure of a five-star result

One weak early review does more damage than a small job is worth. Take these on only when you're 100% confident you'll deliver — then over-deliver. If you're not sure it's a fit, that's a flag, not a bid.

— NON-NEGOTIABLE

Sengar's clients stay with Sengar.

Every client reaches you through the agency, and that's how it stays. **You must never take on or continue work directly with a Sengar client** — no side arrangements, no "let's skip the middle-man," however it's framed or however good the offer sounds. This protects your payment, protects the client, and is the agreement we all work under.



Contacted off Upwork? Tell your manager on Flock — immediately

If a client reaches out on LinkedIn, GitHub, email, WhatsApp or any channel outside the agency, don't engage on the substance. Screenshot it, post it to your manager on Flock right away, and let them steer what happens next.

— THE RULES

Eight habits behind every good message.

1

Reply within 5 minutes, even with nothing to show

"Seen it — reading through now, proper answer by [time]." A holding reply in two minutes beats a complete one in two hours, and this applies after office hours too. Silence reads as "gone".

2

Scope it before you commit

Anything about scope, timeline, or "can you do X" — check the code and the ask before you answer. Don't fire back a delivery promise from a gut guess.

3

Under-promise on time

Estimate honestly, add a buffer, and give a date you'd bet money on. A deadline you beat builds more trust than an ambitious one you miss.

4

Speak outcomes, not internals

The client cares what it does for them, not your stack or your refactor. Translate every "how" into a "what this means for you". (More on this below.)

5

Ask before you assume

Five minutes of sharp questions up front saves days of building the wrong thing. Never start coding on a guess about what they meant.

6**Never go dark**

While you're heads-down, still send a one-line progress note. To a client, no news feels like bad news — a quick "on track, X done, Y next" keeps them calm.

7**Bad news early and straight**

A blocker, a slip, a bug — say it the moment you know, with your plan to fix it. Clients forgive problems. They don't forgive being surprised at the deadline.

8**Money talk isn't yours to handle**

Any price, budget or financial discussion goes to Mohit — never quote a rate or negotiate yourself (see the price card below). And keep the whole conversation on Upwork until a contract is signed.

— HOW THE CONVERSATION RUNS

Six moves that win the conversation.

Fast replies get you into the room. These six keep you there. They're the difference between a chat that fizzles and one that turns into a contract — and none of them need you to sound like a salesperson.

1**Agree first, then move**

Whatever the client says, find the true part and agree with it before you add anything: "You're right", "I'm with you on that", "Makes sense". Treat pushback as a complaint to acknowledge, not an objection to argue with — argue with it and you turn it into one. Agreement is the fastest route to the next step.

2 Research them before you reply

Two minutes on their website, their Upwork history and LinkedIn before your first real answer. They've already looked you up; return the favour. A reply that mentions something specific about their business beats a generic one every time.

3 Ask before you pitch

People hire to solve a problem, so find the problem before describing the solution. Three questions that do the work: "What do you want your next developer to do that your current one doesn't?" · "What's the single most important thing on this one, and why?" · "Why is this coming up now?" Let them talk; they'll tell you how to help them.

4 Find out who else decides

Early and politely: "Who else would you like involved in this decision?" If there's a CTO, co-founder or board, offer a short call with them or a one-pager the client can forward. Never assume the person messaging you is the only one who says yes.

5 Put every estimate and promise in writing

Clients believe what they can see, not what they're told. After a call, send a short written note of what was agreed: scope, dates, next step. It makes you look credible, it protects you, and it's what gets forwarded to the person who signs.

6 Never leave without a return time

"Let me check and get back to you" with no time attached creates doubt and lowers your value in their eyes. Always attach a time: "Back to you by 4pm your time." Then beat it.

How fast, and what "answered" means.

< 5 min

FIRST REPLY · ANY HOUR

The moment a message lands, office hours or not. Even just "on it — details by [time]." Replying in 5 minutes instead of 10 makes a win about 9 times more likely.

Same day

REAL ANSWER OR PLAN

Once you've scoped it properly: a clear answer, or the questions you need answered first.

Daily

WHILE YOU BUILD

A one-line progress note, every working day, until it ships. No news feels like bad news.

— TRANSLATE

Say it the way the client hears it.

Most of what feels normal to say to another dev lands as noise — or worry — to a client. Keep the engineering in your head; send them the meaning.

YOU THINK

"I refactored the auth module and cleaned up the token handling."

YOU SEND

Login is faster now and more secure — you won't notice the change except that it just works.

YOU THINK

"Hit a CORS issue talking to their API."

YOU SEND

Ran into a small config problem connecting the two systems. Fixing it now — your timeline isn't affected.

YOU THINK

"It's basically done, just some edge cases left."

YOU SEND

The main flow works end to end. I'm hardening the last few cases so it doesn't break on odd inputs — done by [date].

YOU THINK

"Their API is undocumented, this is painful."

YOU SEND

Their system isn't well documented, so I'm adding extra checks to be safe. Worth a small delay for something reliable.

— SITUATIONS

What to say when they say...

THE FIRST MESSAGE

CLIENT

"Hi, saw your profile. I need a React dashboard built — can you help?"

✓ SEND

"Hi [name] — yes, this is squarely what I do. Two quick things so I can be useful fast: is there existing code or a design to work from, and what's the rough timeline? Happy to sketch a first step once I know."

✗ AVOID

A bare "yes" — or the opposite, a wall of ten technical questions before you've even said hello.

"HAVE YOU DONE THIS BEFORE?"

CLIENT

"Have you built Stripe billing with multi-tenant before?"

✓ SEND

"Good question — let me pull up the closest thing I've built so I can give you a specific, honest answer rather than a vague yes. Back to you later today."

✗ AVOID

"Yes of course, many times, no problem at all." (*claiming before you've checked*)

VAGUE REQUIREMENTS

CLIENT

"I want to build an app like Uber but for home services."

✓ SEND

"Love the idea. To scope it right: who books first — customers or providers? Web, mobile, or both? And what's the one thing v1 must do? With those three answers I can propose a lean Phase 1 you can launch with."

✗ AVOID

Quoting a timeline on the spot — or silently starting to build what you **assume** they mean.

"HOW LONG WILL IT TAKE?"

CLIENT

"Roughly how long for the whole thing?"

✓ SEND

"Let me scope it properly rather than guess — I'll come back today with a date I'll actually hold to, and what it depends on. Better you get a number you can plan around than an optimistic one."

✗ AVOID

Blurting "a couple of days!" to sound impressive — then living under a deadline you invented.

SCOPE CREEP

CLIENT

"Can you also add payments and push notifications while you're at it?"

✓ SEND

"Both doable. Quick heads-up so there are no surprises: they add roughly [X] to the timeline. Want them in this phase, or shall we ship the core first and add them as a fast follow?"

✗ AVOID

Silently absorbing it (you slip, you resent it) — or a flat "that wasn't in scope" with no path forward.

SOMETHING'S WRONG / YOU'RE LATE

CLIENT

"This isn't quite what I asked for, and it feels behind schedule."

✓ SEND

"Thanks for flagging — that's mine to put right. Give me until [time] to get to the bottom of it, and I'll come back with what happened, the new date, and exactly how I'll fix it."

✗ AVOID

A long defensive explanation — or going quiet and hoping to make it up before the deadline lands.

"SEND ME SOME INFORMATION"

CLIENT

"Can you just send me some info and I'll have a look?"

✓ SEND

"Of course — sending it now while we're chatting. It's a one-pager: how we work, the trial week, and two examples close to yours. What's the one thing you most want it to cover, so I point you to the right part?"

✗ AVOID

"Sure, I'll email it over later this week."
Information requested is sent during the conversation, not after it — once the thread goes cold, the info is read by nobody.

"WE ALREADY WORK WITH SOMEONE"

CLIENT

"Thanks, but we already have a developer / an agency."

✓ SEND

"That's good to hear — a setup that works is worth protecting, and I'm not asking you to switch. If it's useful, I'll stay in touch so you've got a second option ready when a deadline or a skill gap shows up. What would you want a backup team to be strongest at?"

✗ AVOID

Arguing they'd be better off with us, or disappearing. Stay in the conversation, stay positive, and end on a question.

"NOT INTERESTED"

CLIENT

"Not interested, thanks."

✓ SEND

"Fair enough, and thanks for replying. Can I ask what you're sorting out at the moment instead? If it isn't us, I might still know someone."

✗ AVOID

Taking it personally, going quiet, or replying with a longer pitch. Stay flat, agree first, be interested in the "not interested". If they say no a second time, hand the thread to Mohit rather than asking again.

"I NEED TO CHECK WITH MY BOSS / THE TEAM"

CLIENT

"I'll need to run this past my CTO and co-founder."

✓ SEND

"Makes sense — who else will weigh in on it? Happy to join a quick call with all of you, or write a one-pager you can forward so nothing gets lost in translation. Which is easier?"

✗ AVOID

"Sure, let me know!" and then waiting. Find out who actually decides, offer to get in front of them, and keep your energy up — this is normal, not a brush-off.

"CAN WE TRY BEFORE WE COMMIT?"

CLIENT

"Could we do a small test task first before committing?"

✓ SEND

"Yes — that's exactly what our one-week trial is for. Let's start it today: send me the first task and I'll have Mohit set the contract up so everything stays on Upwork and you're covered from minute one. What would you want to see by Friday to call it a win?"

✗ AVOID

Agreeing in principle and leaving it for "next week". A trial agreed is a trial started today — and the contract still goes live before any work (Mohit sets it up).

"LET'S MOVE TO WHATSAPP / EMAIL"

CLIENT

"Easier if we chat on WhatsApp — here's my number."

✓ SEND

"Happy to, once we've got a contract in place. Until then Upwork's own rules mean everything stays here — it keeps you protected and everything tracked, and I promise you won't be left waiting."

✗ AVOID

Jumping off-platform before a contract — against Upwork's rules, and it drops the paper trail that protects everyone.

ANYTHING ABOUT PRICE

CLIENT

"What would you charge for this? Any chance of a better rate?"

✓ SEND

"Happy to make the commercials work for you. Rates and budgets are handled by Mohit directly — let me get him into a chat with you so you get a straight answer from the right person."

✗ AVOID

Naming a number or a discount yourself. Pricing is never ours to move — a guess undercuts Mohit and boxes you in.



Then, internally: the moment price comes up, post a screenshot of the client's message in the Flock group and tag it for Mohit — don't wait, so he can pick up the pricing conversation himself.

— WHEN THE CLIENT IS TECHNICAL

When the client is a developer too.

Sometimes the person hiring can read code as well as you can — a CTO, a technical founder, a senior engineer bringing on extra hands. The relationship shifts: they judge

you on precision, not polish, and they can tell in seconds when an answer is real. This is your home turf — but stay sharp. Lean into some things, drop others.

LEAN INTO

- Real terms and specifics — the actual stack, the actual constraint
- Saying "let me check and confirm" when you're not certain — they respect it more than a smooth yes
- Talking shop as a peer — your real reasoning, out loud
- Trade-offs and reasoning, not just the conclusion

DROP

- Sales gloss and superlatives — "best", "world-class", "3000 developers"
- Bluffing to sound confident — they will catch the gap
- Over-explaining the basics they already know
- Winning the argument instead of the client

A technical client is still the client. Win them with honesty and precision, not by winning the argument — and never claim a certainty you don't have. When you don't know, "let me check and get back to you" is the strong answer, not the weak one.

A TECHNICAL TEST QUESTION

CLIENT

"How would you handle race conditions in the checkout flow?"

✓ SEND

"Good one. I'd handle it with idempotency keys plus a row-level lock on the order, so a double-submit can't double-charge. Happy to walk you through the trade-offs — short version here, or a quick call?"

✗ AVOID

A confident, hand-wavy answer you pieced together on the spot. A technical client spots the gap in seconds — and then doubts everything else you've said.

THEY PUSH AN APPROACH YOU DOUBT

CLIENT

"Just keep the auth token in localStorage, it's simpler."

✓ SEND

"Get the appeal — it is simpler. One flag: localStorage is readable by any script on the page, so for auth tokens I'd normally use httpOnly cookies instead. Happy to go your way — want my two-minute reasoning first?"

✗ AVOID

"No, that's wrong, never do that." — or quietly building it their way while you're sure it'll bite them later.

— A FEW MORE HOUSE RULES

The everyday operating rules.



Record every client call

Meetings, demos, discovery calls — record them (give the client a quick heads-up first, for accurate notes). Share the recording with the team so we can review, learn, and get sharper each time. A call nobody captured is learning lost.



Get the contract started before any work

In every client call, steer the conversation toward getting the contract awarded so work can officially begin. We can't bill without an active contract — so never start building until the client has approved it and it's live. Contract first, then work.



A bonus is paid as a bonus, never as logged hours

If a client wants to pay something extra, it comes through the bonus payment on the Upwork contract. **Never collect a bonus as work hours, and never log time you didn't actually work.** On an hourly contract the time log has to be real work on that contract — padding it breaks Upwork's terms and puts your payment protection and your account at risk. If a client asks, tell them plainly that we can't take it as hours, and that sending it as a bonus on the contract is the way to do it. Then hand the money side to Mohit.



Applying to jobs yourself? Never boost — and track your connects

You're welcome to send your own proposals alongside the ones our team bids for you. Two rules: **never boost a proposal** — boosting stakes extra connects to rank higher and can burn through your balance with no cap. And post a short end-of-day summary of the connects you used, so we can see what's being spent and where it's landing.



Not your skillset? Say so — don't force it

If a request is outside your skills or clearly not a fit, flag it here rather than winging it — we'll match the right person. And any financial or commercial discussion goes to Mohit, not you.

— BEFORE YOU HIT SEND

Five-second gut check.

- ☒ Did I reply within 5 minutes — even after hours, even just to acknowledge?
- ☒ Is every date in here one I'd actually bet on?
- ☒ Would a non-technical person understand this — no jargon?
- ☒ Am I assuming something I could just ask?
- ☒ Did I agree with something before I answered?
- ☒ If I said "let me check", did I give a time I'll be back?

- ☑ Is there any money talk here that should go to Mohit instead?
 - ☑ Would I be happy for a lead to read this exact message?
-

— WHEN TO BRING IN MOHIT

You run the build. You don't run alone.

BRING IN MOHIT

- Any price, budget or financial discussion → hand it to Mohit and drop a screenshot in the Flock group
- A client reaching you off Upwork → screenshot it and tell your manager on Flock, right away
- Work outside your skills, or a request that isn't a fit → flag it so we match the right person
- A client agrees to the trial week, or says no twice → tell Mohit the same day (the trial starts today; a second no is his to handle)
- An unhappy client, a dispute, or scope ballooning — and when you're unsure, ask first

HANDLE IT YOURSELF

- Technical questions — this is your home turf
- Honest time estimates, once you've scoped it
- Clarifying questions to pin down what they need
- Daily progress and any "heads-up" on delays

— IN ONE LINE

Reply in five minutes, talk like a human, and let them trust you before they've seen a line of code.

Team Acknowledgement

I acknowledge that I have read and understood the **Sengar Consultancy Talking to Clients playbook (Upwork etiquette)**.

I agree to follow these standards in every client interaction, and specifically commit to:

- Replying to every client message on Upwork **within 5 minutes**, including after office hours.
- Keeping the Upwork mobile app installed with push notifications on.
- Watching and monitoring my Upwork notifications carefully for at least my **first 3 months** after joining.
- Agreeing first and asking before pitching; putting every estimate and promise in writing; never leaving a client without a time I will be back.
- Telling Mohit the same day when a client agrees to the trial week, or says no twice.
- Fast, honest replies; service before selling; written options; contract-first work; and professional communication that represents Sengar well.

Full name: _____

Signature: _____

Date: _____

Digital edition: sengarconsultancy.org/upwork-etiquette