Moving Goalpost

Facilitation guide · team, one facilitator

What this is: a short, facilitated conversation you run when a customer's requirements have shifted since you started — and it's beginning to feel like a fight about scope. Instead of tracking and defending against "creep," the team surfaces what the customer now understands that they didn't before, makes that learning visible to both sides, and renegotiates from shared understanding. It takes about 30–45 minutes. This is a relationship conversation as much as a delivery one — your tone carries it.

The one line to hold: your job is to change the conversation from "who changed what" to "what have we learnt." The customer usually didn't change the requirement — they understood something new after seeing the work. **The requirement didn't change. The understanding did.**

· WHEN DO I USE THIS?

Use this when

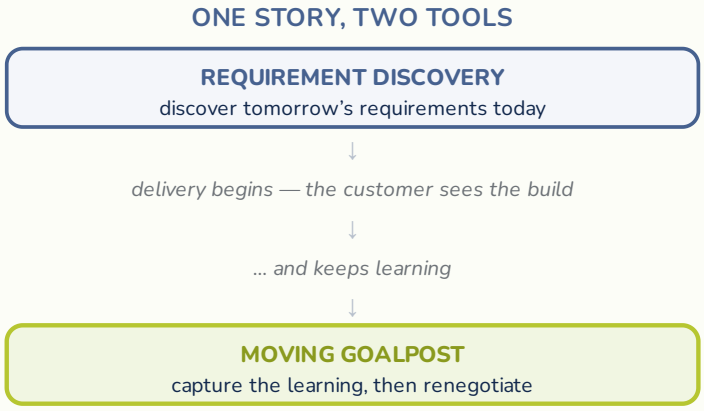
GOOD FIT

A customer's requirements keep changing and it's turning adversarial; you're heading into a "that's not what we agreed" conversation; scope has grown and you need to renegotiate without blame poisoning the relationship; the customer has seen the build and now wants something different. Bring a real, live change — not scope management in the abstract.

WRONG TOOL

You want to surface the real requirement *before* work begins › Requirement Discovery. You're deciding whether a plan should survive a change at all › a different Change Track tool. This one is for a specific, in-flight shift in what the customer understands.

The pairing worth knowing: Moving Goalpost has a natural partner in **Requirement Discovery**. They tell one continuous story about the customer's evolving understanding — one before the build, one in flight.



© Witmer

PART 1 RUNNING THE CONVERSATION

team, one facilitator

1 · WHAT IS THIS REALLY DOING?

The one idea

Customers often don't know what they really need until they see the work. What looks like scope creep is usually new understanding — they're not moving the goalposts; they can finally see the situation clearly. This conversation captures what everyone understood then and understands now, makes the learning visible, and uses it as the starting point for renegotiation.

TWO CONVERSATIONS, SAME CHANGE

OLD

"You're changing scope."

accusation → a fight

NEW

"What have we learnt since we started?"

shared learning → a renegotiation

The requirement didn't change. The understanding did.

2 · WHAT DO I NEED?

What you'll need

A live change to work with (core)	A real, current situation where the customer's requirements have shifted — not a hypothetical. The tool works on one specific change, not scope management in general.
Both understandings, captured honestly (core)	What the customer understood then, and what they understand now — captured privately first, so neither side performs. This honest before-and-after is what the whole tool rests on.
The people from the original conversation	Whoever was part of the first requirement discussion, on both sides. If they can't attend, bring the original notes — don't reconstruct "then" from memory, or you'll get it wrong.
A facilitator who keeps it curious	One person holds the tone: never debating whether the requirement "really" changed, always asking what the customer understands now that they didn't before. The capture sheet carries the then-versus-now and the renegotiation.

Resilience · Change Track · about 30–45 minutes, depending on how far understanding has drifted and how much needs renegotiating. The capture is quick; the renegotiation takes the most time.

© Witmer

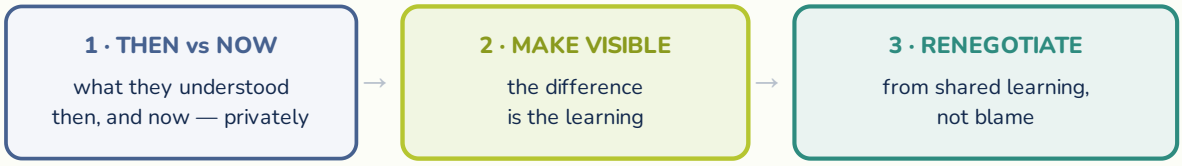
3 · HOW DO I RUN IT?

The run-sheet

Set the tone before anything else: this is "what have we learnt," not "who changed what." Everything downstream depends on the room believing you're here to understand the shift, not assign fault. Capture privately first so no one performs.

1. Frame the conversation	**3 min**	*Say it out loud: this isn't "what did you change," it's "what have we learnt since we started."* Name that you're here to understand the shift, not assign fault.
2. Capture "then" privately	**7 min**	*Each person writes independently what everyone believed the customer needed at the start.* The original shared picture — privately first, so no one performs for the room.
3. Capture "now" privately	**7 min**	*Each person writes independently what everyone now believes the customer needs* — what's become clear since, and what changed it (often: seeing the build).
4. Make the learning visible	**8 min**	*Lay "then" beside "now" and name the difference together as learning.* "Here's what we both understand now that we didn't then." A shift that happened to both of you — not scope one side added.
5. Renegotiate from the learning	**15 min**	*With the shift owned by both sides, renegotiate scope, time or cost from the new shared understanding.* Agree how the plan changes, and record the agreed changes before ending the discussion. This is where the tool pays off — don't stop before it.
6. Confirm the shift is shared	**5 min**	*Check both sides agree on what changed and how the plan now reflects it.* If it still feels like one side conceding, the learning wasn't made visible — go back to step 4.

THEN vs NOW → MAKE VISIBLE → RENEGOTIATE



Understand the learning first. Decide whether to accept it second.

The negotiation is the payoff — don't stop before it.

© Witmer

FACILITATOR WARNING — NEVER DEBATE WHETHER IT CHANGED

The single fastest way to kill this conversation is to let it become an argument about whether the requirement changed. It sounds like this — Customer: “That’s what I always wanted.” Engineering: “No, it wasn’t.” And it’s over. The instant you feel that argument starting, redirect: **don’t ask whether it changed — ask what the customer understands today that they didn’t understand then.** That question is curious; the other is defensive. And hold the other line just as firmly: this tool doesn’t mean accepting every request. Understanding the learning comes first; deciding whether to accept the change comes second. You’re making the negotiation fairer, not skipping it.

PART 2 GETTING IT RIGHT

why it works & common mistakes

4 · WHY DOES THIS WORK?

Why it works

Capturing then versus now makes the shift a fact rather than a fault — and once it’s a fact both sides can see, there’s nothing to defend. The blame that turned the moment into a fight has nowhere to attach, so the conversation can move to the only useful question: given what we both understand now, how should the plan change? The negotiation doesn’t disappear — it just starts from shared ground.

© Witmer

5 · WHAT IF IT GOES WRONG?

Common mistakes

What happened	The fix
It became “did you change it or not?”	The conversation-killer. Redirect immediately: not whether it changed, but what the customer understands now that they didn’t then. Keep it curious.
The team framed the shift as “scope you added”	That’s an accusation, and the tone collapses. The difference is “what we both understand now” — something that happened to both sides, never something one side did.
You captured out loud, together	People perform when others are listening. Capture “then” and “now” privately first, then compare — the honesty is what makes the learning legitimate.
You stopped at “we understand each other now”	Understanding isn’t the output — the renegotiation is. Carry it through to an agreed change, and record it before you finish.
It became a licence to accept everything	The tool makes negotiation fairer, not automatic. Understand the learning first; then decide whether to accept the change. You’re still allowed to say no.
“Then” was reconstructed by the wrong people	If no one from the original discussion is present, “then” becomes a guess. Bring the people who were actually there, or capture it from the original notes.

Before you close the conversation

- ✓ “Then” and “now” were captured privately, by people who were actually there
- ✓ The difference was named as shared learning — not “scope you added”
- ✓ You renegotiated and recorded an agreed change — not just mutual understanding
- ✓ Both sides agree what changed and how the plan now reflects it

© Witmer

6 · HOW DO I MAKE THIS STICK?

Take it back to work

THE HABIT

Next time a requirement “moves,” ask before you defend:

“What have we learnt since we started?”

The more the team treats change as learning rather than betrayal, the less scope discussion feels like conflict. Make “what have we learnt since we started?” your first question whenever requirements appear to change.

Appendix · Reference

You should not need this to start. Read it before, or if a run goes sideways.

7 · A WORKED RUN

Watch a fight become a renegotiation

A client kept “adding” to a dashboard build, and the team was heading into a scope fight. The desired outcome hadn’t changed — but what the client wanted kept shifting.

Stage	What happened
Frame	The lead opened with “let’s understand what’s become clear since we started” — not “let’s talk about the changes.” The room relaxed.
Then vs now	Privately: at kick-off, the client understood they wanted “a dashboard.” Now, having used the early version, they understood they needed to make specific weekly decisions with it.
Make it visible	Laid side by side, the difference was obvious — and clearly learning, not creep. Using the build had taught the client what they actually needed. Nobody had done anything wrong.
Renegotiate	From that shared picture: two features dropped, one added, timeline extended two weeks. A renegotiation, not a fight.
Record	Timeline updated, feature changes documented, both sides confirmed the new agreement — part of the tool, not an afterthought.

© Witmer

8 · WHERE DOES THIS FIT?

Where this sits in the Gym

The real need	Use this
Renegotiate a mid-flight change from shared learning, not blame	**This tool — Moving Goalpost.** Surface what the customer now understands, make it visible, renegotiate.
Surface the real requirement *before* work begins	**Requirement Discovery** — helps customers discover latent needs up front, by reaction. Moving Goalpost handles the learning that still emerges once they see the solution. One before, one in flight.

Within Change Track, the three tools answer different questions:

The real need	Use this
Decide whether the current plan should survive a change	**Pivot Point**
Build a new plan after the old one is no longer viable	**Replan the Plan**
Renegotiate changing customer understanding	**Moving Goalpost** (this tool)

The move underneath: when a requirement "moves," treat it as the customer learning something — not as bad behaviour. Make the learning visible so both sides own it, then renegotiate from there. Together with Requirement Discovery, this forms one philosophy of the customer's evolving understanding: discover what they'll need before you build, and understand what they've learnt once they see it.

© Witmer