Replan the Plan

Facilitation guide · team, one facilitator

What this is: a facilitated rebuild session you run when reality has overtaken your plan and you've already decided it has to change. Instead of patching the old plan again, the team treats it as dead, folds it away, and rebuilds from today's facts — then, and only then, brings the old plan back to recover anything still genuinely useful. Rebuild first; recover afterwards. It takes about 60–90 minutes depending on the size of the plan.

The one line to hold: your job is to stop the team revising and make them rebuild. The whole mechanism lives in one rule — no one looks at the old plan until the new one's first version exists. **Every patch keeps part of a world that no longer exists.**

· WHEN DO I USE THIS?

Use this when

GOOD FIT

The ground has shifted — scope, team, timeline or goal — and the plan no longer fits; you've patched it so many times no one fully trusts it; you keep working *around* the plan instead of *from* it. You've already decided the plan has to change and need to rebuild, not just tweak.

WRONG TOOL

You're still deciding *whether* to change course › Pivot Point. You need to define *which problem* to solve › Future Back. You just need to adjust pace or rhythm mid-flight › Mid-Sprint Reset. This tool assumes the plan is already dead.

The boundary that matters most: this is the second half of a sequence. **Pivot Point** decides *whether* the plan survives; **Replan the Plan** assumes it's already dead and builds the replacement. If you haven't yet decided the plan must change, you're still in Pivot Point — come back here once the answer is "no, it can't survive."

THE CHANGE-TRACK SEQUENCE



© Witmer

PART 1 RUNNING THE SESSION

team, one facilitator

1 · WHAT IS THIS REALLY DOING?

The one idea

When reality changes, the original plan quietly becomes an anchor — the reference point you revise *against* instead of starting fresh. That's not emotional attachment — it's simply the brain's tendency to keep working from what already exists, and it's why revisions pile up as Version 8 of Version 7 of Version 6 instead of Version 1 of reality. Every patch preserves assumptions built for a world that no longer exists — assumptions you can't see precisely because you never rebuilt without them. This tool removes that anchor: fold the plan away so it can't be revised against, rebuild from today's facts, and only then reconcile.

TWO WAYS TO MEET A CHANGED REALITY

PATCH *revise against the old plan*	V6 → V7 → V8 → V9… *each inherits yesterday's assumptions*	✗
REBUILD *build from today's facts*	**V1 of reality** *no dead assumptions carried forward*	✓

You're not adjusting a plan — you're defending an obsolete one.

You are not adjusting a plan — you are defending an obsolete one.

© Witmer

2 · WHAT DO I NEED?

What you'll need

The right people (core)	Bring the people who understand today's reality and can shape the new plan. Don't run this with only reviewers or approvers — the people who build the plan have to be in the room.
A plan you've already decided to change (core)	This tool rebuilds; it doesn't decide whether to rebuild. Come having concluded the current plan no longer fits. If you're still deciding, that's Pivot Point.
Today's facts (core)	The current state, the current constraints and the current goal — the real ones now, not the ones the old plan assumed. This is what you rebuild from.
A facilitator	Holds the one hard rule: the old plan is folded away and no one looks at it until the new plan's first version exists. Without this, the team drifts back to patching.
The capture sheet (supplied)	Carries the rebuild — today's facts and the new plan — and the reconciliation, so detachment doesn't lose real work.

Resilience · Change Track · about 60–90 minutes, depending on the size of the plan and how much has changed. Most of the time is the rebuild; reconciliation is quicker.

3 · HOW DO I RUN IT?

The run-sheet

Open with three words: this plan is dead. Not "let's update it," not "let's revise it" — dead. That framing is what forces the rebuild; everything else in the run-sheet protects it.

Before you begin — three ticks, thirty seconds

- ✓ Everyone agrees the current plan no longer fits
- ✓ Everyone accepts today's facts
- ✓ Everyone agrees no one will look at the old plan until reconciliation

1. Declare the plan dead	**2 min**	*Say it plainly: "this plan is dead."* Name what changed in reality that killed it, so the whole team agrees it's genuinely obsolete — not just tired.
2. Fold it away	**1 min**	*Physically remove the old plan from view — fold the paper, close the document, minimise the screen.* The small physical act reinforces letting go. No one looks at it again until Step 5.

© Witmer

3. Write today's facts	**10 min**	*Capture reality as it actually is now — today's facts:* current state, current constraints, current goal. Be honest about what's changed; this is the ground the new plan stands on.
4. Rebuild from scratch	**30 min**	*Build the plan you'd write if this were day one and no plan existed.* Work only from today's facts. Don't reconstruct the old plan from memory — that's patching in disguise.
5. Reconcile with the old plan	**15 min**	*Now — and only now — bring the old plan back.* Recover anything genuinely still valid; consciously leave behind the assumptions you were only carrying out of habit.
6. Check it starts from today	**2 min**	*Confirm the rebuilt plan begins with today's goal, today's constraints and today's reality.* If you can't point to all three, you've rebuilt yesterday's plan — go back to Step 4.

DETACH → REBUILD → RECONCILE

1 · DETACH
declare it dead,
fold it away

→

2 · REBUILD
from today's facts,
as if day one

→

3 · RECONCILE
bring the old plan back,
recover what's valid

No one looks at the old plan until Version 1 is complete.

Rebuild first. Recover what's still useful afterwards.

FACILITATOR TIP — GUARD THE NO-PEEKING RULE

The whole mechanism rests on one instruction, and it will be tested within minutes: someone will say "let me just check what we had for the timeline." Don't allow it. One glance at the old plan re-anchors the room, and you're revising again without noticing. Hold the line: nobody looks until Version 1 of the new plan exists. If the team is anxious about losing good work, reassure them with the sequence — you *will* bring the old plan back at Step 5 and recover everything still valid. Rebuild first, recover afterwards. If someone insists they need the old plan, remind them you'll bring it back during reconciliation — nothing useful will be lost. Letting go of the old plan isn't reckless; it's what makes the good work visible again.

© Witmer

PART 2 **GETTING IT RIGHT** why it works & common mistakes

4 · WHY DOES THIS WORK?

Why it works

When reality changes, the original plan becomes an anchor — a reference point you revise against instead of starting fresh. That's a cognitive default, not attachment, so every patch preserves assumptions you can no longer see. Folding the plan away removes the anchor: you can't revise against something you can't look at, so you're forced to build from today's facts. The old assumptions become visible by their absence — you notice what's missing only when you rebuild without it. Reconciling afterwards keeps the detachment honest, so you recover real work instead of discarding it.

5 · WHAT IF IT GOES WRONG?

Common mistakes

What happened	The fix
Someone peeked at the old plan	One glance re-anchors the room. Hold the rule hard: nobody looks until Version 1 exists. If it happened, name it and re-start the rebuild clean.
The team rebuilt the old plan from memory	Reconstructing the old plan is patching in disguise. Pull them back to today's facts: "if no plan had ever existed, what would we build now?"
You softened "dead" to "update"	The framing does the work. "Update" invites a patch; "dead" forces a rebuild. Keep the word.
You skipped reconciliation	Detachment without reconciliation throws away real work — and makes people resist the tool next time. Always bring the old plan back at the end.
The "new" plan still assumed yesterday	Run the check: does it start from today's goal, constraints and reality? If not, you rebuilt yesterday's plan with new formatting.
The team kept debating whether the plan was dead	That decision belongs in Pivot Point, not here. Pause the session; resume only once everyone agrees the old plan no longer fits.
You used it before deciding to change	This tool rebuilds; it doesn't decide. If the team is still debating whether to change course, that's Pivot Point — come back once the plan is genuinely dead.

© Witmer

Before you close the session

- ✓ The old plan stayed folded away until Version 1 of the new plan existed
- ✓ The new plan was built from today's facts, not reconstructed from memory
- ✓ You reconciled — recovered what's still valid, left the dead assumptions behind
- ✓ The plan starts from today's goal, today's constraints and today's reality

6 · HOW DO I MAKE THIS STICK?

Take it back to work

THE HABIT

Next time reality moves, ask before you revise:

"Am I patching a dead plan?"

Soon you'll catch yourself asking "am I patching a dead plan?" before you revise anything. You start reaching to rebuild from today rather than revise from yesterday, and "this plan is dead" becomes a phrase the team says for itself.

Appendix · Reference

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

7 · A WORKED RUN

Watch a plan get rebuilt

A team's six-month delivery plan had been overtaken: the scope had halved, two people had left, and the deadline had moved in. It had been patched weekly and no one trusted it.

Stage	What happened
Declare & fold	The lead said "this plan is dead" and named why: half the scope and two people gone. The document was closed — not reopened until reconciliation.
Today's facts	Current state: four people, halved scope, eight weeks. Current goal: ship the core, not the original everything. The reality, written plainly.
Rebuild	Built fresh: a two-phase plan around the smaller team and tighter deadline. The old plan would never have produced this structure — it was anchored to a larger team and broader scope.
Reconcile	Old plan reopened. Two things recovered: a risk register still valid, and a test approach worth keeping. The rest — built for the old scope — left behind.
Check	Started from today's goal (core ship), today's constraints (four people, eight weeks), today's reality (halved scope). All three — a genuine rebuild.

© Witmer

8 · WHERE DOES THIS FIT?

Where this sits in the Gym

The real need	Use this
Rebuild an execution plan after reality has overtaken it	**This tool — Replan the Plan.** Treat the plan as dead, rebuild from today, reconcile afterwards.
Decide *whether* the current plan should survive at all	**Pivot Point** — the trigger that decides. This is the rebuild once the answer is "no." The two form a sequence.
Define *which problem* you should be solving	**Future Back** — problem definition from a future outcome. This rebuilds the *plan*, not the problem.
Adjust pace or rhythm mid-flight	**Mid-Sprint Reset** — adjusts an in-flight plan. This rebuilds a whole plan after a reality shift. Adjust vs rebuild.

The move underneath: when the world changes, remove the old plan from view so it can't anchor you, rebuild from what's actually true now, and only then recover what still holds. That same move — detach, rebuild, reconcile — works wherever yesterday's thinking is quietly shaping today's, and you need Version 1 of reality instead of Version 8 of a memory.

© Witmer