Outcome Definition System

Definition guide · define exactly what “done” looks like before you start — so the work can finish, not drift

Your job: you’re about to start a piece of work, and before you begin you’re going to draw its finish line — concretely enough that anyone could tell whether you’ve crossed it. Write what “done” must produce, make every item answerable yes or no, then lock it before work begins. This is your work to do as the owner, at kickoff, usually on your own — a focused fifteen minutes before the work starts, not a discussion partway through.

The one line: if you can’t say what finished looks like, you can’t finish — you can only stop.

THE RULE THAT MAKES THIS WORK

Every criterion must be able to **fail** a test before it can pass one. If you can’t answer **yes** or **no**, it isn’t a definition of done — it’s just a goal. That single test, applied to every line, is the whole mechanism. The question you keep asking is: **“how would someone prove this is true?”**

· WHEN DO I USE THIS?

Use this when

GOOD FIT

- You’re starting work whose **“done” is a fuzzy assumption** right now
- Similar work in the past drifted and never cleanly finished
- Several people would define “finished” differently for this work
- The scope feels like it could quietly expand forever

USE ANOTHER TOOL

- The work is finished and you just need it **visible** to whoever needs it → **Done ≡ Show It** (the quick check at the end)
- You need the **order** the work happens in → **Execution Sequencing System**
- You’re partway through and “done” has already moved → that’s scope change — name it, don’t redraw the line quietly

The test question: “Am I about to start work whose finish line isn’t drawn yet?” If yes, draw it here first. This tool defines done at the *start*; Done ≡ Show It confirms it’s shown at the *finish*.

© Witmer

PART 1 DEFINE & TEST

before the work begins

1 · WHAT IS THIS REALLY DOING?

The one idea

THE ONE IDEA

An undefined “done” is a finish line that keeps moving (drifts) — everyone fills the gap in differently, and nothing can be held against it. Write done as concrete, testable criteria and the line is fixed in place — which is the difference between being able to **finish** and merely being able to **stop**. The test is what does the work: a completion criterion (a yes/no check) you can’t actually answer yes or no on is one you can quietly renegotiate later — so it’s not really part of “done” at all.

THE SAME “DONE”, TWO WAYS

A GOAL (can’t be tested)

“the onboarding works well”

can’t answer yes or no

→ the finish line moves

you can only stop, never finish

A CRITERION (testable)

“new users reach first value in under 3 minutes”

answers yes or no

→ the finish line holds

you can prove you’re done

If it can’t be answered yes or no, it isn’t a definition of done.

2 · WHAT DO I NEED?

What you’ll need

Item	What it is, and when it’s used
The proof question (the heart of it)	For every criterion: *“how would someone prove this is true?”* If there’s no yes/no answer, it isn’t a criterion yet. Everything else exists to serve this question.
One piece of work, at the start	Something you’re about to begin and are accountable for finishing. Define done before you start, not partway through.
The definition sheet (core)	Where you write each done-criterion and its yes/no test. It forces the testability — a criterion with no test doesn’t count.
Honesty about vagueness	Don’t let “works well” or “high quality” stand as criteria. If you can’t test it, it isn’t part of “done”.

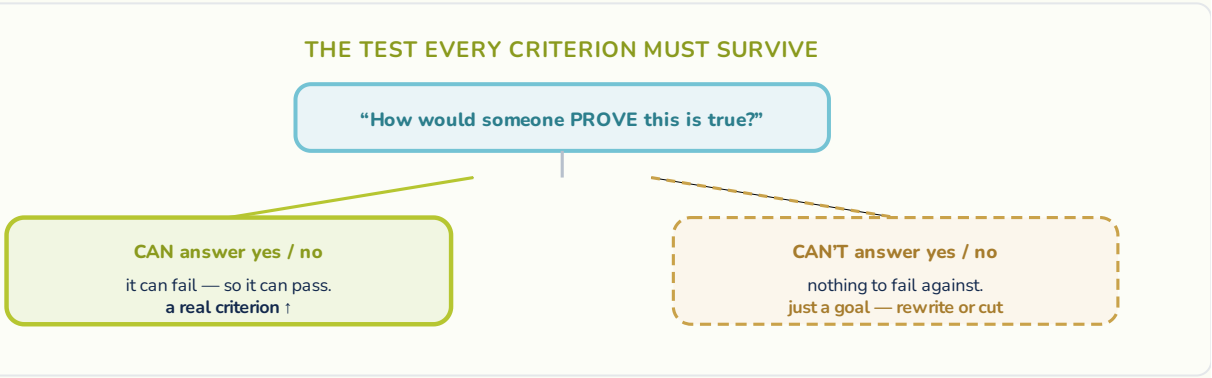
© Witmer

3 · HOW DO I RUN IT?

Define, then test

Two moves: get the criteria down, then make every one testable. The second move is where the tool really works — don't skip it.

1 · Name the work & a first version of done	**2 min**	Write what you're making and what "finished" currently means to you. *The fuzzy version is fine here — you're going to sharpen it.*
2 · List the done-criteria	**6 min**	Write the specific things that must exist or be true to finish — *deliverables, conditions, completion checks.* Aim for a handful, not a wish-list.
3 · Make each one testable	**6 min**	For every criterion ask: *"how would someone prove this is true?"* Rewrite each until it answers yes or no. Cut or fix anything vague — "works well", "polished" are not criteria.



PART 2 LOCK & FINISH

the fixed line, and the end

4 · HOW DO I LOCK IT?

Fix the finish line

The tested set is now the definition of done. Agree it before work begins — with whoever you answer to, if the work has a client or a manager. From here it's fixed: **the work is finished only when every criterion can be answered "yes"** — no sooner. If the line genuinely needs to change later, that's scope change: name it out loud and re-agree it, never move it quietly.

© Witmer

5 · WHAT IF IT GOES WRONG?

Common mistakes

What you'll see	The fix
Criteria that sound good but can't be tested	The main trap. "Works well", "high quality", "complete", "polished", "customer ready" — none answer yes or no. They're goals, not criteria. Ask "how would someone prove this?" and rewrite until there's a yes/no answer.
A wish-list of thirty items	Done isn't everything you could do — it's the line that means finished. A handful of real criteria beats a page of hopeful ones.
Defining done after you've started	A line drawn once the work's underway has already moved. Draw it at kickoff, before the first piece of work.
Moving the finish line once work has started	That's the drift the tool exists to stop. If the line must change, call it scope change and re-agree it — don't slide it.
Confusing stopping with finishing	Running out of time is stopping. Finishing is every criterion answering yes. The definition is how you tell them apart.

Before you start the work, check your definition:

- ✓ Every criterion can be answered yes or no — you tested each one
- ✓ No vague words survived — no "works well", "high quality", "polished"
- ✓ It's a handful of real criteria, not a wish-list
- ✓ Someone else could hold the work up and judge it without you
- ✓ It's agreed and fixed before the first piece of work

6 · HOW DO I MAKE THIS STICK?

Take it back to work

THE HABIT

Before starting anything that matters, ask one question: "how will I know this is finished — in a way someone else could check?" If you can't answer it, you haven't defined done yet — and the work will drift until you do.

The distinction worth keeping: **stopping is not finishing**. Stopping is running out of time or patience; finishing is the moment every criterion answers yes. Once you've felt the difference, working without a definition of done starts to feel like working blind.

© Witmer

Appendix a worked example · where this sits

7 · A WORKED EXAMPLE

Watch a vague goal become testable

The work: a team is asked to "improve the onboarding flow". As written, it could never end — there's always something more to improve.

The goal (vague)	The criterion (testable, yes/no)
"Onboarding works well"	New users reach first value in under 3 minutes. **(yes / no)**
"Sign-up is simple"	The sign-up form has four fields or fewer. **(yes / no)**
"Fewer people drop off"	Drop-off between steps two and three is below 15%. **(yes / no)**
"It's polished"	*Cut — can't be tested. If it matters, name the specific check; otherwise it isn't part of done.*

The lesson, not stated but felt: "improve the onboarding" had no end — it was a direction, not a finish line. The moment each part became a yes/no check, the work had somewhere to stop that actually meant finished.

8 · WHERE DOES THIS FIT?

Where this sits in the Gym

Four Focus tools run a piece of work from start to finish — each owns a different moment, with almost no overlap:

Tool	The question it answers
Outcome Definition System	**What are we trying to finish?** — the testable definition of done. (This tool, at kickoff.)
Execution Sequencing System	**What's the best path?** — the order the work is done in.
Decision-Driven Meetings	**How do we steer it?** — the meetings that keep the work moving.
Done ≡ Show It	**Is it actually visible as finished?** — the 30-second check at the very end.

The clean pairing: define done thoroughly here at the start; at the finish, use **Done ≡ Show It** to confirm the finished work is visible to whoever needs it. Same journey, two ends.

© Witmer