Execution Queue Control

Facilitator guide

Your job: count what's really in progress. Set a limit below that number. Make something finish before anything new starts. Then agree the rule for what happens when new work arrives. The hard part isn't the number — it's setting it low enough to force a stop. A note on the term: a **work-in-progress limit** (shortened to **WIP limit** from here) is just the most things the team allows itself to have started at once.

What is Execution Queue Control? A team feels busy all day, yet little gets finished — because everything is started at once and nothing is protected to completion. This activity gets the team to count what's really in flight, cap it below that number, and agree one rule for new work. You run it; the outcome is a WIP limit the team owns and an intake rule it will hold.

THE ONE THING TO REMEMBER

The queue already exists. You're only deciding whether it's visible or hidden.

If the team remembers nothing else in six months, it should be this. Every new "yes" is already delaying something — this tool just makes that cost visible now, so the team can control it.

WHEN DO I USE THIS?

The right moment

GOOD FIT

Everything is "in progress" and nothing finishes. The team keeps starting new work, is pulled between tasks, and delivery has slowed to a crawl — even though no one is idle.

WRONG TOOL

Work has stalled on one blocker › Dependency Snapshot. Time is leaking inside one flow › Constraint Snapshot. You're defending a deadline and nothing's overloaded yet › Protect the Critical Path.

The one test before you start: is the problem *too much work started at once* (this tool), or *one thing blocked* (Dependency Snapshot)? This is for overload — too much in flight — not a single stuck item.

© Witmer

· WHAT WILL THE TEAM LEAVE WITH?

What you'll leave with

WHAT YOU GET

A WIP limit the team agreed, set below the count. One intake rule — new work starts only when a slot frees. And a clear picture of what's being finished first, and what waits.

WHY IT MATTERS

The team sees its hidden queue for the first time — and realises it can control it. Saying "yes" stops being without cost: it now visibly means choosing what waits. The team finishes more by starting less.

PART 1 BEFORE THE ACTIVITY

get ready

1 · WHAT IS MY JOB HERE?

Your job

You are running this activity to one outcome: a WIP limit the team owns, set below current load, and an intake rule it will hold. Your job is to:

- get the team to **count** what's really in progress — usually more than anyone expected
- hold the line on a limit set **below** that count, so it forces a finish before a start
- make the excess **visible** in a waiting line — parked, not cancelled
- force the intake rule **concrete** — "we'll be mindful" is not a rule

2 · WHO DOES WHAT?

Roles

Who	What they do
You (facilitator)	Count the work in progress, hold the limit below that count, run the limit tests, drive the excess into a visible waiting line, and force the intake rule concrete. You do **not** accept the comfortable number.
Participants (a team)	List what's in progress, count it honestly, agree the limit and the intake rule, and decide what finishes first and what waits.

Say this to the team before you start (in your own words): "We're going to put up everything we've started but not finished, count it, and set a deliberate limit — below that number — so something has to finish before we start more. Then we'll agree what happens when new work arrives. It'll feel restrictive; that's the point." Don't assume anyone read the app page.

© Witmer

3 · HOW DO I START?

Opening script — 2 minutes

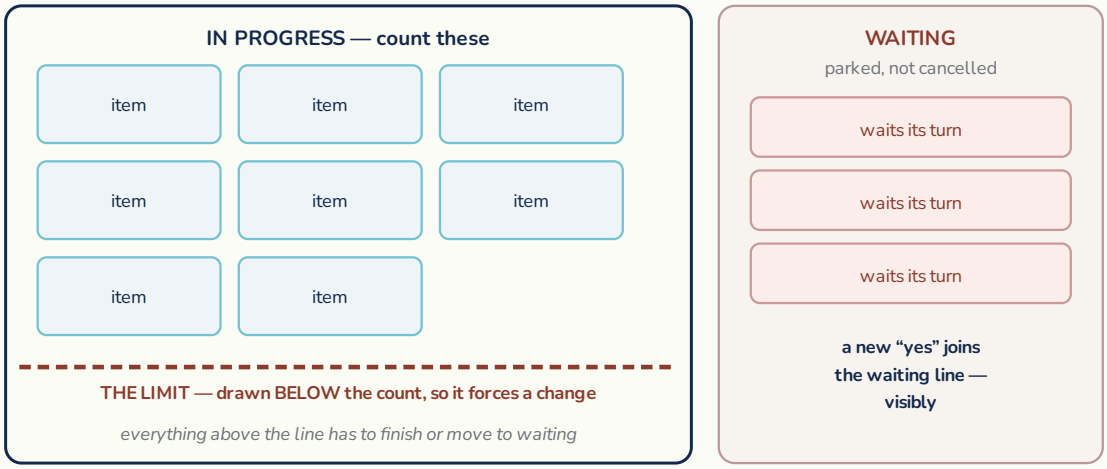
Say this in your own words before you start. Do not read the app page out loud.

> SAY
>
> “Right now it feels like we’re busy all day and somehow nothing gets finished. That’s usually because we’ve started far more than we can carry at once. So first we’ll count everything that’s actually in progress — it’s usually a shock. Then we’ll set a limit below that number, on purpose, so something has to finish before we start the next thing. And we’ll agree one rule for when new work lands. Nothing gets cancelled today — things wait, visibly, in the open.”

4 · HOW DO I PREPARE?

Setup and materials

ONE BOARD — WHAT’S IN PROGRESS, AND WHAT WAITS



One board, two zones: the **in-progress** column you count, and a **waiting line** beside it. The **limit** is a line drawn *below* the count — everything above it must finish or move to waiting. Setup is a board, sticky notes and a marker; under two minutes.

What	Why it's there
Your real in-progress work	Everything currently started-but-not-finished — from the board, tracker or what the team knows. If little is in progress, there's no overload to fix. This is the critical ingredient, and the hardest thing to count honestly.
A board or wall	An "in progress" column you can count, and room for a waiting line beside it.
Sticky notes + marker	One note per item in progress; the marker writes the limit and the intake rule where the team will see them.

Everything is simple — lay it out before the team arrives. Board up, an in-progress column and a waiting line marked, notes and a marker to hand. Setup should take under two minutes. About 25–30 minutes to run, a team and one facilitator.

Words to know: in progress is anything started and not finished. The **WIP limit** is the most items allowed in progress at once. The **waiting line** is where parked work sits — not cancelled. The **intake rule** is what happens when new work arrives.

© Witmer

PART 2 DURING THE ACTIVITY

run it

Before you begin — 30-second check

- ✓ There is real overload — more is in progress than the team can clear (not one stuck item)
- ✓ The board has an in-progress column to count and room for a waiting line beside it
- ✓ The team understands nothing gets cancelled today — things wait, visibly
- ✓ You are ready to hold the line on a limit **below** the count, not at it

5 · HOW DOES THE LIMIT WORK?

The limit that forces a change — the make-or-break move

This is where the tool lives. A limit set at or above current load changes nothing — the team just keeps going. The value is a number low enough to force a finish before a start. Don't rely on judgement; run the three tests.

THE LIMIT ONLY WORKS IF IT FORCES A CHANGE

LIMIT AT / ABOVE THE COUNT	LIMIT BELOW THE COUNT
Count 8 · "cap it at 8"	**Count 8 · limit 3**
nothing has to stop — the team just keeps going. Changes nothing.	five items must finish or wait — it forces a finish before a start.

Run the three tests on the number the team proposes:

1 · COUNT	2 · BELOW	3 · STOP
"How many are in progress right now?"	"Is the limit below that count?"	"Must something finish before a new start?"

If any answer is "no", the limit is too high — lower it until at least one current item has to wait.

THE ONE RULE

Set the limit below today's count. The number only works if it's below the current count and makes something finish first. If the proposed limit doesn't make at least one current item wait, it's too high — lower it until it forces a change.

6 · WHAT DO I DO NOW?

Run the activity

Count in progress › Set limit below it › Excess to waiting › Agree the intake rule

© Witmer

1. Count what's in progress 4 min	"Put up everything that's started but not finished — one note each." Then count them out loud. The number is usually higher than anyone expected.
2. Set the limit — below that number 9 min	"What's the most we should have in progress at once?" Agree a number below today's count, so it forces something to stop. **Run the three tests (§5).** Count › Below › Stop. If the number doesn't make at least one current item wait, it's too high. This is where you add the most — don't accept the comfortable number.
3. Move the excess to a waiting line 5 min	"We're over the limit — which items move to 'waiting'?" Park the excess in a visible queue. They're not cancelled; they wait their turn. **Feeling uncomfortable means the tool is working.** Don't soften it — that visible queue is what makes the cost of over-starting real.
4. Agree the intake rule 6 min	"When new work arrives, what happens?" Write the rule: new work waits until something finishes; to start it now, name what stops. Photograph the board before anyone leaves. **Force it concrete.** "We'll be mindful" is not a rule. The test: next time someone brings urgent work, does the rule tell the team what to do without a debate?

These times are indicative and will depend on the nature of the problem and the size of the team. Counting is quick; step 2 (setting it below the count) is where the real work is. Step 4 (the intake rule) is what gets skipped — protect it. A limit with no intake rule erodes by next week.

While you run — when to step in, when to stay out · a principle, not a step

STEP IN

The limit lands at or above current load → push it lower.

They won't move anything to waiting → the limit isn't real.

The intake rule is vague → force it concrete.

They argue the count → just count what's started.

STAY OUT

Parking started work feels uncomfortable → that's the point.

They debate the exact number → below the count is what matters.

Someone resists their item going to waiting → let the limit decide.

The board looks bare after parking → good, that's focus.

© Witmer

The clearest tell it's time to step in: the team sets the limit at exactly today's count — "we'll just not start more." That changes nothing. Push: "what has to finish before anyone starts the next thing?" The limit must force a stop.

PART 3 AFTER THE ACTIVITY

make it stick

7 · HOW DO I FINISH?

Closing script — 2 minutes

SAY

"Look at the board. We had eight things going and thought that was normal — now we've got a limit of three and a waiting line we can see. Nothing was cancelled; the rest just waits, in the open. From now on, when new work lands, we don't quietly squeeze it in — we ask what finishes or waits. The queue was always there. Now it's visible, and we control it instead of it controlling us."

8 · DID WE GET A GOOD RESULT?

Check your run

DONE WELL

A limit set below current load, the excess parked in a visible waiting line, and a concrete intake rule. The team can say: "to start something now, we name what stops."

A HOLLOW RUN

A limit at or above current load, nothing moved to waiting, and a vague "we'll be mindful." If so, nothing was actually limited — go back to the count and set it lower.

WHAT LEAVES THE ROOM

WIP limit: 3 items in progress

Counted today: 8 open — set below that, on purpose, so it forces a stop.

a number the team set, below current load

THE INTAKE RULE

New work waits until something finishes. To start something now, name what stops.

a rule that makes a new "yes" cost something — visibly

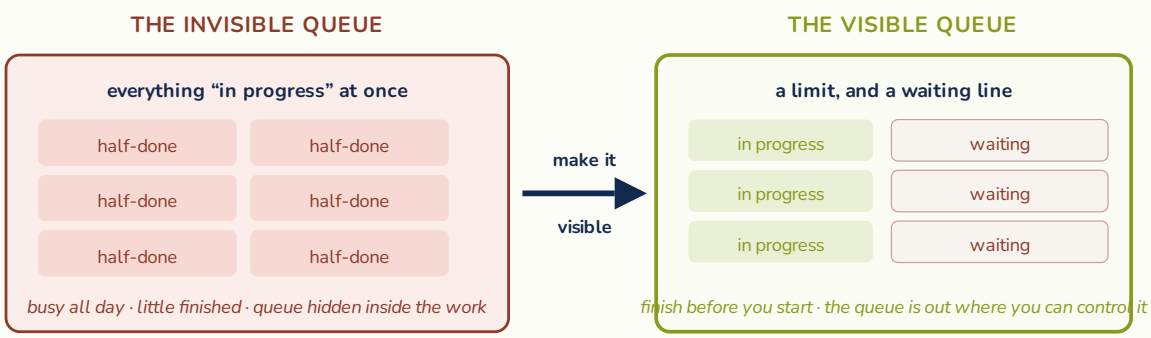
A number below current load, and a rule that holds when the next request arrives.

9 · WHY DOES THIS WORK?

The idea underneath — the invisible queue

Every time the team says yes to new work, it's deciding what gets delayed — but each new "yes" feels free, because the cost lands later, on someone else's item. The queue is already there; it's just hidden inside half-finished work. A limit set below the count makes that queue visible, where the team can control it.

© Witmer



The queue already exists. You’re only deciding whether it’s visible or hidden.

A hidden queue controls you. A visible one, you control.

Before people leave — the finish check

- ✓ The team counted what’s really in progress — out loud, honestly
- ✓ The limit is set **below** that count, so something must finish first
- ✓ The excess is parked in a visible waiting line — not cancelled
- ✓ A concrete intake rule is written: to start now, name what stops
- ✓ The limit and rule are somewhere the team will see them at the next request
- ✓ The board has been photographed before anyone leaves

10 · HOW DO I MAKE THIS A HABIT?

Build the habit — 30 seconds

THE HABIT

“Before saying yes to new work: what now waits? Every “yes” delays something — name what.”

The lasting change is that “yes” stops being without cost. A team that has run this once tends to ask, at the next request, what its “yes” will cost — and to make that cost visible on the board rather than hidden in the work.

© Witmer

Appendix · Reference

You should not need this while you run the activity. Read it before, or use it if something goes wrong.

11 · A WORKED RUN

Watch the facilitator move

A team had “about five or six” things going. Watch the moment the limit became real.

Team “We could cap it at six — that’s about what we run.”
Facilitator “How many are actually in progress right now? Count them.”
Team “…eight, actually. Maybe nine with the bugfix.”
Facilitator “So a limit of six still lets you keep all of them. What number forces something to finish first?”
Team “…three. Three would mean we actually finish things before starting more.”

What the facilitator does: (moves five items to a waiting line) “Then it’s three. The rest wait — and new work waits for a finish. Which five move first?” That last question makes the trade-off immediate: the team has to name, out loud, what stops. They didn’t accept the comfortable number; they made the team count, then set the limit below it. Six would have changed nothing; three forced the finish.

12 · WHAT IF SOMETHING GOES WRONG?

Recovery guide

What happens	What to do
The team sets the limit at current load	It forces nothing. Make them count first, then set it below — run the Below and Stop tests until the number forces a stop.
“We can’t stop any of these — all urgent”	Then nothing finishes and everything’s late. Park the least-urgent in waiting; the limit reveals that “all urgent” isn’t survivable.
Work types are very different (bugs vs features)	Limit by lane if needed, but keep it simple — even one shared number beats none. Don’t over-engineer it.
A genuine emergency exceeds the limit	Allowed — but it bumps something to waiting, visibly. The rule is “to start this, what stops?”, not “never exceed”.
A stakeholder says “this one’s an exception”	The most common way a limit dies. An exception is allowed — but it must **visibly move another item to waiting**. If nothing moves, the limit has silently increased and you’re back to overload. Make the swap happen on the board, in front of everyone.
The team accepts work anyway (permission)	See the watch-for below — back the rule with whoever can say no upward, or it won’t hold.

© Witmer

“This is just a kanban WIP limit”	It is, deliberately. The value is agreeing the number and the intake rule together, and seeing the cost of “yes” on the board. If the team already holds a real limit, it doesn’t need this.

Watch for: the team knows, but accepts work anyway. Sometimes the limit is clear and the team breaks it anyway — that’s a permission problem, not a visibility one. The intake rule has to be backed by whoever can say no upward. Name who that is, or the limit erodes the first time pressure arrives.

13 · WHERE DOES THIS FIT?

Where this sits in the Gym

Four tools for four different flow problems. Reach for the one that matches what’s actually happening.

If the question is…	Use this
“Too much work is in progress at once”	**Execution Queue Control** (this tool) — cap what’s in flight so work finishes before more starts.
“Where is time disappearing?” (in one flow)	**Constraint Snapshot** — find the stall where work sits waiting.
“What are we waiting for?” (work already stuck)	**Dependency Snapshot** — trace the stuck work to its source blocker.
“What must not be interrupted?” (before the cycle)	**Protect the Critical Path** — protect the pacing work before it’s at risk.

Rule of thumb: Execution Queue Control limits what gets in; Constraint Snapshot finds where time leaks once it’s in. Intake versus flow.

© Witmer