Capacity & Feasibility System

Facilitator guide

Your job: give the team a fixed pool of capacity tokens, have it spend them on the commitments it has already taken on — guessing the cost — then turn over the real, hidden costs so the team discovers it is already over capacity, and has to take wanted work off the table. The tokens running out do the teaching; you count real capacity, hold the hidden costs, and never let the team borrow.

The one idea: you can't say no to a real list with an imaginary limit. The team committed before it knew the cost. This kit turns the guess into tokens that run out — and the real costs, hidden until the reveal, almost always add up to more than the team has. Count capacity before you commit.

· WHEN DO I USE THIS?

Use this when

GOOD FIT

The team is about to take on a batch of work — a quarter, a release, a roadmap — and wants to know what it can really afford, before it commits. Every planning round ends with more committed than it can deliver, and the true cost of each piece is unclear until you're deep in it.

WRONG TOOL

You're already overloaded and must cut fast › Capacity Triage. You want to feel why more work doesn't mean more output › Load vs Throughput Lab. The same crunch keeps recurring and you want to know why › Crunch Cycle Review.

The one test before you start: is this a real set of commitments the team is taking on — and has it not finally committed yet? If the team is already overloaded and just needs to cut, that's Capacity Triage. This is the check before the yes.

· WHAT WILL THE TEAM LEAVE WITH?

What you'll leave with

WHAT YOU GET

A clear picture of real capacity and how much is already spoken for, the true cost of each commitment out in the open, and an affordable set the team can actually deliver — with every dropped item handled on purpose.

WHY IT MATTERS

The team feels the pool run out. The real cost was bigger than anyone guessed; it was already over before adding anything new. Saying no gets easier once you can see the tokens run out — and the count holds against the next request.

© Witmer

PART 1 BEFORE THE ACTIVITY

get ready

1 · WHAT IS MY JOB HERE?

Your job

You are running this to one outcome: an affordable set of commitments under a real, counted capacity. Your job is to:

- count **real** capacity — the availability the team will actually have, not the best case
- set and **hold the hidden costs** — keep them face down until the reveal; the surprise is the whole tool
- run the spend and the reveal, then protect the **drop-to-fit** — the step that gets squeezed
- never let the team **borrow** — when the pool's empty, it's empty

THE ONE IDEA

Count capacity before you commit. The commitment is concrete and visible; its capacity cost stays hidden. Teams overcommit not because they're greedy but because they say yes to a real list against an imaginary limit. Turn capacity into a pool that runs out, turn the hidden costs over, and the gap stops being a guess — feasibility becomes a decision, not a discovery.

2 · WHO DOES WHAT?

Roles

Role	What they do
You (facilitator)	Count the capacity honestly, set and hold the hidden costs, run the spend and the reveal, keep time, hold the no-borrowing rule. Stay neutral on which work survives — it's cost versus capacity, not your favourites.
Participants (3–6)	The people who own the work. They write the commitments, spend the tokens, face the reveal, and decide what to drop. 45–90 minutes.

3 · HOW DO I OPEN?

Opening script — 2–3 minutes

In your own words, never read aloud. Keep the real costs hidden — that surprise is the whole tool.

SAY — IN YOUR OWN WORDS

"Our real capacity is a fixed pool of tokens — that's all we have. We'll spend it on the work we've already taken on, guessing what each costs. Then we'll turn over what each really costs. Most teams find they're already over — before adding anything new. The point isn't to feel bad; it's to see what we can actually afford, and leave with a count we can hold against the next request."

© Witmer

4 · HOW DO I PREPARE THE SPACE?

Setup and materials

THE BOARD — A FIXED POOL, SPENT ONTO COMMITMENTS

CAPACITY — the pool



sky = free to spend grey = in flight

Count it honestly. Set aside in-flight first.

What's left is all you have to spend.

no borrowing — when the pool's empty, it's empty

COMMITMENTS — the cards

Commitment card
cost hidden under a flap

Commitment card
cost hidden under a flap

Commitment card
cost hidden under a flap

Commitment card
cost hidden under a flap

Team spends guessed tokens onto cards — costs stay face down *until the reveal.*

Component	What it is, and when you use it
Capacity tokens	One pool for the team; one token = a unit of real capacity (e.g. half a person-week). Step 2: you count the team's true capacity into tokens. This is all they have to spend.
"In flight" markers	A few tokens in a different colour for work already underway. Step 3: set these aside first, so free capacity is what's left.
Commitment cards (hidden cost)	One card per commitment, with its real capacity cost concealed under a flap or on the back. Step 4: laid out; Step 6: costs revealed. The concealment is the surprise — you set the costs, the team never sees them early.
The board + capture space	Two areas — capacity (the pool) and commitments (the cards); tokens travel from pool to cards as the team spends. Capture space records the affordable set and what was dropped.

PART 2 DURING THE ACTIVITY

run it

Before you begin — 30-second check

- ✓ This is a real set of commitments the team is taking on — and it hasn't finally committed yet
- ✓ The people who own the work are in the room (3–6)
- ✓ You've set each card's hidden cost — a little above what a busy team would guess
- ✓ The board's ready: capacity pool one side, commitments the other

© Witmer

5 · HOW DO I RUN IT?

How to run it

Brief the team (2–3 min), then run these in order. The bold part is what you say or do; the note under it is where you add the most. Hold the timings.

1. Set up	**5 min**	*Board out: capacity area + commitments area.* You hold the hidden costs. Appoint the facilitator up front.
2. Count real capacity	**10 min**	*"What's our true capacity?"* Count it into tokens. Count the capacity you'll *actually* have — not the capacity you hope you'll have. This is all they have.
3. Set aside work in flight	**5 min**	*Remove tokens for current work (grey).* Make them see how little is really free before any new commitment is even discussed.
4. Lay out commitments	**10 min**	*One card per commitment.* The team writes them; the real cost stays hidden. Don't reveal costs yet.
5. Spend — guess the cost	**10–15 min**	*Move guessed tokens onto each kept card.* Let them spend freely; don't hint at real costs. Once tokens are placed on a commitment, don't let them be moved until every commitment has been estimated — or the reveal loses its force.
6. Turn over the real costs	**5 min**	*Reveal each hidden cost; add them up.* This is the moment — let it land. Don't rush to fix.
7. See you're over	**5 min**	*True costs exceed the pool.* You are already over. Don't fix it yet — sit with it. The quiet is the lesson landing.
8. Drop to fit	**10–20 min**	*Remove commitments until costs fit.* Ask: which commitment gives us the least value for the capacity it consumes? And: if we'd known the real cost before planning began, would we still have committed to this? Each on purpose: keep / push / resource / drop.
9. Commit	**5 min**	*Lock the affordable set; photograph it.* Keep the capacity count — it's the limit you hold against the next request.

These times are indicative and will depend on the nature of the problem and the size of the team; the whole activity runs 45–90 minutes. The step that overruns is counting capacity (step 2); the step that gets squeezed is dropping to fit (step 8) — protect it. An undecided overcommitment is exactly the problem the team came to prevent.

© Witmer

6 · WHAT DOES THE REVEAL LOOK LIKE?

The reveal — the heart of the tool

The guessed tokens were optimistic. When the hidden costs turn over and are added up, they almost always exceed the pool — the team is already over, before anything new. Let the room go quiet before you move on.



The guesses were optimistic. Added up, the real costs need more than the pool holds — before anything new is added. Let it land before you fix it.

When to step in — and when to stay out · a principle, not a step

STEP IN

Capacity is counted at best case → "what's really free, after what's already on us?"

Someone wants to borrow or invent tokens → hold the pool; the limit is the point.

A dropped item is left vague → force keep / push / resource / drop.

They treat it as picking favourites → bring it back to cost versus capacity.

STAY OUT

The room goes quiet at the reveal → that's the lesson landing.

They argue over a guessed cost → the real cost is coming; let it ride.

Someone's reluctant to drop wanted work → the reluctance is the point.

Two commitments compete for the last tokens → let them decide.

The clearest tell it's time to step in: the team starts reasoning "we'll just absorb it" about an over-capacity commitment. That's the old habit — imagining capacity that isn't there. Point at the empty pool: "which tokens pay for it?"

© Witmer

PART 3 AFTER THE ACTIVITY

make it stick

7 · HOW DO I CLOSE?

Closing script — 2 minutes

Point the team at the count it now holds — the thing it carries out of the room.

SAY — IN YOUR OWN WORDS

"We started sure all of this would fit. The reveal showed the real costs needed *[N]* tokens and we had *[M]* — already over, before anything new. So we dropped to fit, on purpose: *[what stayed, what was pushed, resourced, dropped]*. Here's the count we can afford — that's the limit we hold against the next 'can you just…' request. Feasibility is a decision, not a discovery."

8 · DID WE GET A GOOD RESULT?

What good looks like

A GOOD RUN

Capacity is counted honestly, the hidden costs are revealed, the team faces being over, and drops to fit — each cut made on purpose. Costs seen · over-capacity faced · dropped on purpose.

A HOLLOW RUN

Costs are never turned over, the team keeps everything, and leaves as overcommitted as it arrived. If it looks like this, reveal the real costs and make the pool run out — the pool emptying teaches the lesson.

9 · HOW DO THE DROP DECISIONS WORK?

Drop to fit — on purpose

Once you're over, every over-capacity commitment is handled explicitly — never left to drift. Four ways out, each a deliberate choice:

DROP TO FIT — EACH OVER-CAPACITY ITEM HANDLED ON PURPOSE

KEEP	PUSH	RESOURCE	DROP
fits within real capacity	to a later cycle — not now	add real capacity — funded, not wished	not taken on — the no is on record

Ask: which commitment gives us the least value for the capacity it consumes?

Every over-capacity item ends with one of these — on purpose, not by default.

Scope reduction is a new card, not a shortcut. If the team says "we'll keep all seven but shrink each", don't allow it as a vague trim — a reduced-scope commitment only counts if it becomes a clearly defined new card with its own real cost. Otherwise the exercise quietly reintroduces the imaginary capacity it exists to remove.

© Witmer

Before you finish — the check

- ✓ Real capacity was counted honestly, and work in flight set aside first
- ✓ The hidden costs stayed hidden until the reveal — the surprise held
- ✓ The team saw it was already over — the true costs against the pool
- ✓ Every over-capacity item ended keep / push / resource / drop — on purpose
- ✓ The hidden costs weren't negotiated down — the commitments changed instead
- ✓ The affordable set is locked, photographed, and the capacity count kept for next time

10 · HOW DO I MAKE THIS A HABIT?

Take it back to work

THE HABIT

Before the team says yes: "what will this really cost us, and can we afford it?"

Feasibility is a decision made before committing, not a discovery made halfway through. Count capacity before you commit.

Appendix · Reference

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

11 · A WORKED RUN

Watch the reveal land

An eight-person platform team setting its quarter — seven projects, each with a sponsor. The mood is "ambitious but doable."

WORKED RUN — AN EIGHT-PERSON PLATFORM TEAM, SEVEN PROJECTS

COUNT + SET ASIDE

Real capacity counted, then in-flight migration and one person on leave set aside — the free pool is thinner than expected: 8 tokens.

THE REVEAL

Two projects cost nearly double the guess. True cost needs 10 tokens; the team has 8. Already over — before adding anything new.

THE DECISION

Two fit and stay · one pushed · one resourced with a contractor · one dropped. Committed against a count they can point to.

The reveal: the hidden costs turn over one by one. Two projects cost nearly double what the team guessed. Added up, the true costs need ten tokens; the team has eight — already over, before adding anything new. One sponsor says: "I'd already promised mine. I never knew what it actually cost us." A good facilitator prompt here: "which of these commitments would never have entered the plan if we'd seen these costs on day one?" Two projects fit and stay; one is pushed, one is given a contractor (real capacity added), one is dropped. The quarter is committed against a token count the team can now point to.

© Witmer

12 · WHAT GOES WRONG MOST OFTEN?

Common mistakes

Mistake	Why it fails the run
Counting capacity at best case	The pool ends up too big and nothing runs out. Count real availability — after leave, in-flight work, and the rest.
Revealing costs too early	The surprise is the tool. Keep every cost face down until the whole spend is done.
Negotiating the hidden costs down	That's the old habit sneaking back. Don't reduce a revealed cost — change the commitment instead.
Letting the team borrow	Borrowed capacity is the imaginary capacity that caused the overcommitment. Hold the pool.
Ending on an undecided overcommitment	A found gap with no drops is just a diagram. Drop at least to the pool's limit before anyone leaves.

© Witmer

13 · WHAT IF SOMETHING GOES WRONG?

Edge cases

What happens	What to do
The team wants to borrow or invent tokens	Hold the pool. The fixed limit is the whole point — borrowed capacity is the imaginary capacity that caused the overcommitment. Point at the empty pool: "which tokens pay for it?"
Real costs fit the pool with room to spare	Rare. Check the capacity count wasn't at best case and the costs weren't set too low. If it's genuine, the team can afford the batch — commit it and keep the count for next time.
The team refuses to drop anything	Don't fix it for them. Sit with the gap: "the costs are real and the pool is full — what comes off?" The reluctance is the lesson; the decision is theirs.
A sponsor insists their work stays	Keep it about cost versus capacity, not seniority: "it stays — so name what comes off to pay for it." Every kept card must be funded from the same pool.
A guessed cost was already close to real	Fine — the reveal still adds up across all cards. The lesson is the total against the pool, not any single guess.
Someone proposes adding more people midway	Additional people only create new capacity tokens if the capacity is real — genuinely available and funded, not hypothetical. Add tokens only after the team agrees the capacity is real. Wishful capacity is the imaginary capacity the tool exists to expose.
Time runs out before dropping to fit	Never end on an undecided overcommitment. Drop at least to the pool's limit now, even roughly, and mark any close calls to confirm later. A decided-enough set beats an open one.

14 · WHERE DOES THIS FIT?

Where this sits in the Gym

If instead...	Use this
you're already overloaded and must cut fast, now	**Capacity Triage** — the quick cut when you're over already.
you want to feel why more work doesn't mean more output	**Load vs Throughput Lab** — shows how overload slows delivery.
the same crunch keeps recurring and you want to know why	**Crunch Cycle Review** — diagnoses the repeating pattern.

> **Rule of thumb:** if you've not committed yet and want to know what you can afford, this is the tool — it's the check before the yes. If you're already overloaded and just need to cut, that's Capacity Triage.

© Witmer