Crunch Cycle Review

Facilitation guide · team, one facilitator

What this is: a facilitated activity you run when a team keeps living through the same end-of-cycle crunch. It rebuilds the crunch as a line of decisions in time order — a row of dominoes — then tips the first and lets the rest fall, so the team sees the crunch started weeks before the deadline. It ends by naming the earliest decision the team could move next cycle to break the chain. It takes about 45–90 minutes and needs a large table or wall to build the chain on. Your job is to keep the room looking for the earliest link, not the last one.

The one line to hold: the crunch did not begin at the deadline. Keep pushing the team earlier — to the first reasonable-looking decision that started the chain. The instinct is to argue about the final week; the value is in the earliest decision. Keep asking one question — **"What made this necessary?"** — until you reach it. **The crunch began weeks earlier — with one reasonable decision.**

WHEN DO I USE THIS?

Use this when

GOOD FIT

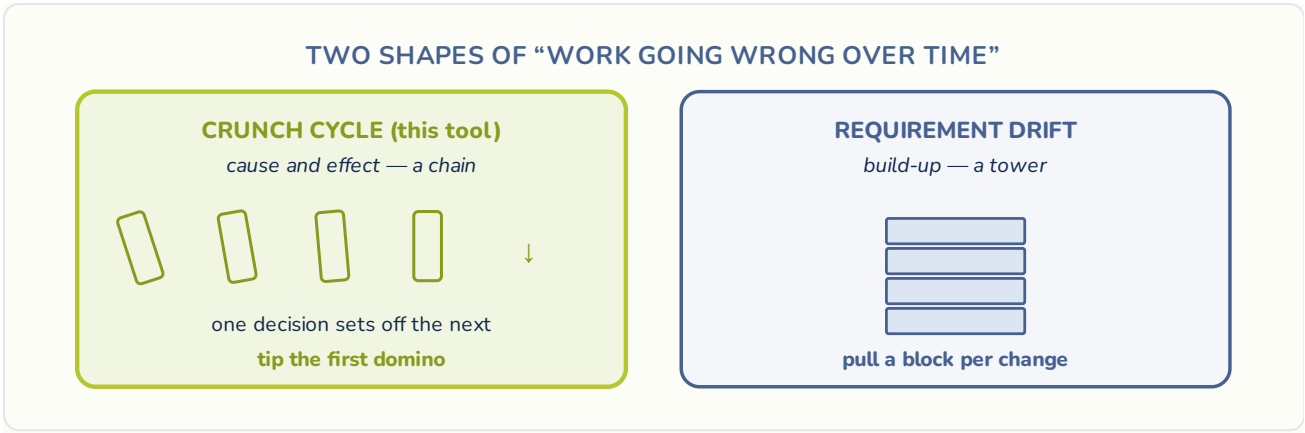
A team that keeps hitting the same end-of-cycle crunch — the late scramble, the long hours every release — and treats each one as a one-off. There's a real, recent crunch to rebuild. The one test: has this crunch happened before, and will it happen again unless something changes? "This is the third release running" means this is the tool.

WRONG TOOL

Many small changes piling up over time › Requirement Drift Simulator. A team that won't stop gathering before deciding › Analysis Paralysis Lab. A single live setback to recover from now › Delivery Recovery Path. A genuine one-off emergency belongs elsewhere — this is for the cycle, not the single incident.

The boundary that matters most: this is not Requirement Drift. That tool is *build-up* — many separate changes pile up, and you pull a block per change. This is *cause and effect* — each decision set off the next, so tipping the first makes the whole thing fall. If the decisions are separate and just piled up, you're running Requirement Drift.

© Witmer



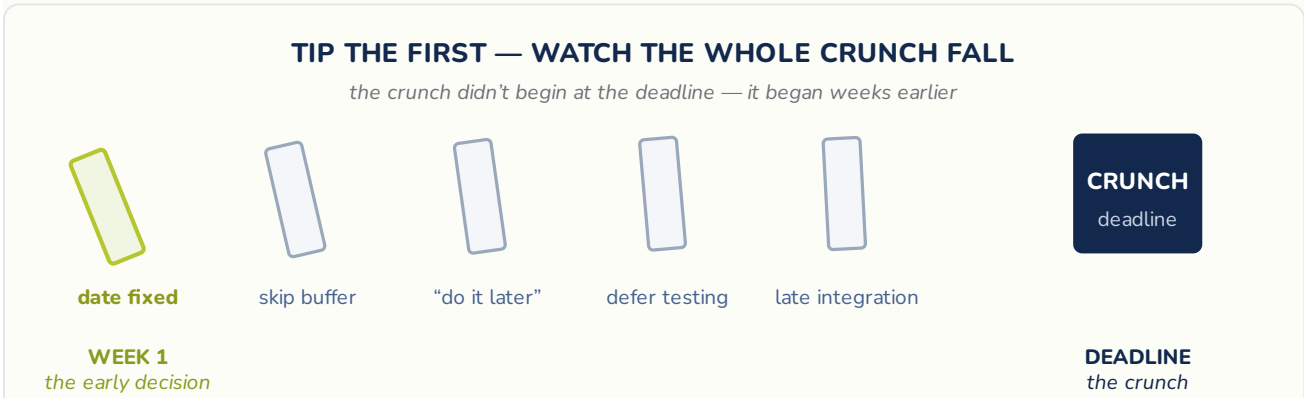
PART 1 RUNNING THE ACTIVITY

team, one facilitator

1 · WHAT IS THIS REALLY DOING?

The one idea

A recurring crunch is the visible end of an invisible chain of decisions. Each decision looked reasonable on its own, so no one saw the chain until the crunch arrived — and the team fixes the last week while the cycle repeats. Laying the chain out and tipping the first domino makes the chain impossible to ignore: the crunch began early, with one reasonable decision, not at the deadline. The fix isn’t working harder at the end — it’s moving the earliest decision.



2 · WHAT DO I NEED?

What you'll need

One recurring crunch (core)	A real, recent crunch the team keeps repeating. They must see it as a pattern, not bad luck.
A large table or clear wall space (core)	The chain is a physical layout, built left to right. You need room to lay the whole thing out so every link is visible.
A line of dominoes	One domino per real decision in the crunch, laid in time order. Tipping the first shows the rest fall.
Write-on decision cards + marker pens	Write-on cards to name each real decision before it's placed, so the chain is the team's actual story.
One facilitator	Runs the flow, keeps the room looking for the earliest decision, and tips the dominoes. Doesn't let it turn into a search for who to blame.

Constraints · Manage Load · 45–90 minutes. This tool needs room — give the chain-building time.

3 · HOW DO I RUN IT?

The run-sheet

Brief the team, then run these in order. The bold part is what you say or do; the move under it is where you earn your keep. Protect Step 2 (rebuilding the chain) and Step 5 (the intervention) — those are where the value lives.

1. Name the crunch	**5 min**	*"Which crunch do we keep living through?"* Pick one real, recent, recurring crunch. Write it as the end-state: "the last two weeks before the X release."
2. Rebuild the chain backwards	**15 min**	*Work backwards one decision at a time. Ask "What made this necessary?" — then ask it again of the answer — until you reach a decision made early in the cycle.* That one question is your rhythm for the whole activity. The team can always see one step back. Write each on a card, then lay them left-to-right, earliest first.
3. Lay the dominoes in time order	**8 min**	*Place one domino per card, left (earliest) to right (the crunch), with a small gap between each so every link is visible.* Read the chain aloud: "this led to this, which led to this…" If two don't connect, a link is missing — ask "what happened between these?"

© Witmer

4. Tip the first — watch it fall	**8 min**	*"Watch what happens when we tip the first decision." Tip it.* The chain reaction runs all the way to the crunch. Let it land in silence, then: "each decision made the next far more likely — the crunch began here, six weeks before anyone felt it."
5. Name the intervention and commit	**12 min**	*"Next cycle, which decision do we move, and to when?"* Name the earliest decision that's a real choice, the change, the owner, and the trigger — the moment next cycle when this decision gets made differently. Commit one, and record the new decision before ending the session.

THE BREAK-POINT TEST — FIND THE EARLIEST DECISION TO MOVE

Not every domino can be moved. Test each one, earliest first, against four questions — the first domino that passes all four is where the team acts next cycle:

- **Was this a choice, not a given?** A fixed external deadline isn't breakable; a self-imposed one is.
- **Could it have been made differently with what we knew then?** Hindsight doesn't count — only what was knowable at the time.
- **Is it early in the chain?** The earlier the decision, the more of the chain it prevents.
- **Would changing it have stopped the next domino?** Mentally remove it — does the chain still fall?

Pick the earliest decision that is a real choice and would have stopped the next one. And make it a **trigger, not an intention**: "scope before we commit a date" is an intention; "at intake, before any date is promised, we scope first — owned by the lead" is a trigger.

PART 2 GETTING IT RIGHT

why it works, redirects & mistakes

4 · WHY DOES THIS WORK?

Why it works & what to watch for

The last week is loud, so everyone tries to fix it. The first week is quiet, so nobody notices it — and the cycle repeats. Laying the chain in time and tipping the first link makes the cause and effect visible: a reasonable early choice pushed the whole chain toward the crunch. Watch for the insight arriving: the team keeps jumping to the final week and you keep pulling them earlier; someone says "that seemed reasonable at the time"; someone says "we've done this every release" — the pattern, named. The talk shifts from blame ("who caused this?") to cause ("what led to what?"). That shift is the insight arriving — let it run, then move to the earliest decision.

© Witmer

5 · WHAT WILL THE TEAM SAY?

What they say — and your move

These are the lines that pull the room to the wrong link. Each has a one-sentence redirect.

What they say	Your response
“The real problem was the last week.”	“That’s where it hurt — but what made the last week unavoidable? Keep going back.”
“We just underestimated.”	“Maybe — but which decision locked in the estimate before we knew enough?”
“It was X’s fault.”	“We’re not finding who — we’re finding when. Which decision started the chain?”
“The deadline was fixed, nothing we could do.”	“Was it truly external, or did we accept it? An accepted date is a choice.”
“It wasn’t a chain — just lots of factors.”	“Not inevitable — but each link made the next far more likely. If we removed this decision, would the next still have happened?”

© Witmer

6 · WHAT IF IT GOES WRONG?

Common mistakes

What happened	The fix
Looking for who to blame	Blame ends the analysis. Redirect to timing, not fault: “we’re finding when the chain started, not who.”
Starting the tip in the middle	The whole reveal is that the *earliest* decision started the fall. Push back to the first link, not the last visible problem.
Picking a late decision to move	Easy to see, but too late to prevent the crunch. The earlier the decision, the more of the chain it stops.
Ending on “more buffer / try harder”	A wish, not a moved decision with a trigger. End with a specific decision attached to a moment in the next cycle.
Treating a truly external deadline as the fix	Find the decision the team actually controls. Usually a “given” was actually accepted — that’s the earliest real choice.
Jumping to solutions before the chain is complete	Finish rebuilding the whole chain first. Fixes proposed halfway through are aimed at the wrong link — don’t jump to solutions until every domino is placed.
Stopping at the first obvious cause	Keep going until the team agrees the chain couldn’t have started earlier. Shallow analysis picks a comfortable middle link.

Before you close the activity

- ✓ The chain is a clean line of cause and effect — each domino leads to the next
- ✓ You tipped the first, not the middle — and the fall landed
- ✓ The team found the earliest decision that was a real choice
- ✓ You left with one change attached to a trigger in the next cycle

© Witmer

7 · HOW DO I MAKE THIS STICK?

Take it back to work

THE HABIT

When a crunch hits, don't ask "how do we survive it?" — ask:

"Which decision, weeks ago, set this up?"

Move that decision earlier next cycle. The more the team hunts the earliest link rather than the last week, the sooner the recurring crunch stops being something to survive and starts being something to prevent.

Appendix · Reference

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

8 · A WORKED RUN

Watch the facilitator move

The crunch: "the last two weeks before every release — long nights, weekends, panic. Third release running."

Stage	What happened
Rebuild backwards	"Crunch ‹ late integration ‹ testing deferred ‹ 'we'll parallelise later' ‹ buffer skipped ‹ date fixed before scoping." Six dominoes.
Tip the first	Tip "date fixed before scoping." The chain runs all the way to the crunch.
The facilitator move	The team starts arguing about the integration week. Redirect: "that domino only fell because the one before it did. What's the first one that was a real choice?" They land on the fixed date.
The intervention	"Next cycle: scope before any date is promised. Owner: lead. Trigger: at intake." The earliest link, moved — and the chain never starts.
Record	Trigger documented, owner confirmed, added to the next release planning checklist — so the agreement survives the session.

© Witmer

EDGE CASES

When a run goes sideways

What happens	What to do
The chain has gaps / links don't connect	"What happened between these two?" A missing link means a hidden decision — surface it before tipping.
Every domino feels external / unavoidable	Test each: "was this a choice we made, or truly given?" Usually at least one "given" was actually accepted — that's the earliest decision.
People disagree about where the chain starts	Build both versions until they converge — the disagreement usually resolves as the earlier links appear. Don't vote too early.
Two chains, not one	Run the most painful, recurring one fully; note the second for a later session. One clear chain beats two half-built.
No recurring pattern — a true one-off	This may be the wrong tool — a single setback is Delivery Recovery Path. This is for the cycle, not the single incident.

WHERE DOES THIS FIT?

Where this sits in the Gym

The shape of the problem	Use this
The same crunch keeps recurring — one decision sets off the next	**This tool — Crunch Cycle Review.** Cause and effect over time. Tip the first domino.
Many small changes build up until the work is unstable	**Requirement Drift Simulator** — scope creep made felt. Pull, don't stack. Build-up, not a chain.
A team that won't stop gathering before deciding	**Analysis Paralysis Lab** — the block is deciding, not a decision chain.
A single live setback to recover from right now	**Delivery Recovery Path** — steady the position, find the next move. One incident, not a cycle.

The move underneath: when the same crunch keeps recurring, stop treating each one as bad luck — lay the decisions out in time, tip the first, and find the earliest one you actually control. That move — from surviving the last week to moving the earliest decision — is what turns a recurring crunch from a fact of life into something you can design out.

© Witmer