

Estimation & Sizing Guide

Formats: Markdown (canonical) | Word (DOCX) | PDF **Updated:** 2026-03-22 **License:** CC BY 4.0 --
Kate Makrigiannis / k8mak.com

A coaching guide for teaching teams how to estimate work using relative sizing -- not hours, not days, not absolute time. Covers the concepts, the scales, and Planning Poker as a facilitation technique.

When to use this

Your team is new to agile estimation, or they're estimating in hours/days and the numbers are consistently wrong. This guide teaches relative sizing from scratch -- why it works, which scales to use, and how to run a Planning Poker session. Pair it with the [Velocity & Forecasting Guide](#) to turn estimates into delivery predictions.

Core concept: Relative sizing

Humans are bad at estimating absolute effort. We consistently underestimate complex work and overestimate simple work. But we're remarkably good at comparing things.

The cats analogy:

Imagine someone asks you: "How much does a house cat weigh?" You'd probably guess wrong. Now imagine someone puts three cats in front of you and asks: "Which one is biggest?" You'd get it right instantly.

That's relative sizing. You're not measuring -- you're comparing. Instead of asking "how many hours will this take?" you ask "is this story bigger, smaller, or about the same as that one?"

Why relative beats absolute:

- Comparing is faster than calculating
- Teams converge on relative size more easily than absolute time
- Relative estimates don't carry the false precision of "this will take 3.5 days"
- They account for uncertainty naturally -- a "big" story acknowledges that you don't know all the details yet

Scales

Pick one scale and use it consistently. Don't switch between scales mid-project.

Fibonacci scale (1, 2, 3, 5, 8, 13, 21)

The most common scale. The gaps get bigger as numbers increase, which reflects reality: the difference between a 1-point and a 2-point story is small, but the difference between an 8 and a 13 is significant.

Points	What it means
1	Trivial. A config change, a copy update, something you could do in your sleep.
2	Small. Clear scope, no unknowns. A straightforward bug fix or a well-defined UI change.
3	Medium-small. Some complexity but well-understood. Might involve 2-3 files or a known pattern.
5	Medium. Real work with some decisions to make. Might need a design review or a small spike.
8	Large. Multiple moving parts, some unknowns. Worth discussing whether it should be split.
13	Very large. Significant unknowns or complexity. Almost certainly should be split into smaller stories.
21	Epic-sized. Too big to estimate meaningfully. Split it before estimating.

Rule of thumb: If a story is bigger than 8, split it. If it's a 21, it's not a story -- it's a project.

T-shirt scale (XS, S, M, L, XL)

Simpler and less precise. Good for teams that are new to estimation or for high-level roadmap planning where precision isn't the point.

Size	Roughly equivalent to
XS	1 point. Done in a few hours.
S	2-3 points. Done in a day.
M	5 points. A couple of days of focused work.
L	8 points. Most of a sprint for one person.
XL	13+ points. Too big -- split it.

When to use T-shirts vs. Fibonacci:

- T-shirts for roadmap-level planning, early discovery, or teams that freeze up with numbers
 - Fibonacci for sprint planning, backlog grooming, and velocity tracking (numbers make math easier)
-

Planning Poker: How to estimate as a team

Planning Poker is a structured way to get the whole team to estimate together. It surfaces disagreements early and prevents anchoring (where one person's estimate influences everyone else).

Setup

- Each team member gets a set of cards with the Fibonacci numbers (1, 2, 3, 5, 8, 13, 21, ?)
- The "?" card means "I don't understand this story well enough to estimate"
- Digital tools work fine: PlanningPoker.com, Miro voting, or just holding up fingers

How to run it

For each story:

1. **The PM reads the story aloud.** Title, description, acceptance criteria. Keep it to 60 seconds.
2. **Quick clarifying questions.** 2 minutes max. Questions about scope, not debates about approach. "Does this include the email notification?" Yes/no. Move on.
3. **Everyone votes simultaneously.** Each person selects their estimate card and reveals at the same time. No peeking, no "what are you going to say?"
4. **If estimates converge (within 1-2 points of each other):** Take the majority or the average. Move to the next story.
5. **If estimates diverge (3+ points apart):** The highest and lowest estimators explain their reasoning. 90 seconds each. Then re-vote. If still divergent after 2 rounds, take the higher estimate and move on.

What divergence tells you

When the team is 3+ points apart, something valuable is happening. Common reasons:

- **Different assumptions about scope.** One person assumed the story includes error handling; another assumed it doesn't. Clarify the scope and re-vote.

- **Different technical knowledge.** The frontend engineer sees a simple UI change; the backend engineer knows it requires a database migration. Listen to the person with domain knowledge.
- **Hidden complexity.** The story seems simple but has edge cases nobody noticed. The divergence is the team surfacing risk.

Don't rush past divergence. It's the most valuable part of Planning Poker. A 2-minute conversation about why estimates differ saves days of rework later.

What NOT to do

These are common traps. Every one of them erodes trust in the estimation process.

DO NOT estimate in time. "This will take 3 days" is not an estimate -- it's a promise. And it's probably wrong. Relative sizing avoids the false precision of time-based estimates.

DO NOT compare teams. Team A's "5-point story" is not the same as Team B's. Points are relative within a team. Cross-team comparison is meaningless and creates toxic competition.

DO NOT treat estimates as commitments. Estimates are forecasts, not contracts. If leadership uses estimates to hold teams accountable for delivery dates, the team will inflate estimates to protect themselves -- and the numbers become useless.

DO NOT estimate alone. One person estimating is just guessing. The whole point of Planning Poker is to combine perspectives and surface hidden complexity.

DO NOT spend more than 2 minutes per story. If a story takes longer than 2 minutes to estimate, it's a sign the story isn't well-defined. Defer it, add it to a grooming session, or ask the PM to clarify before the next planning session.

Worked example

Team: Compass Squad estimating stories for Sprint 7 using Fibonacci scale.

Story	Votes (round 1)	Discussion	Final Estimate
Add "forgot password" link to login screen	1, 1, 2, 1, 1	Converged. Quick change.	1

Story	Votes (round 1)	Discussion	Final Estimate
Implement care plan PDF export	3, 5, 5, 3, 5	Converged. Majority at 5.	5
Add real-time vitals from wearable devices	8, 13, 8, 21, 8	Divergent. The 21 was from the backend engineer who flagged Bluetooth pairing complexity. After discussion, team agreed the story needs splitting.	Split into 3 stories
Dark mode for settings page	2, 2, 3, 2, 2	Converged.	2
AI-powered medication interaction checker	?, 13, ?, 13, ?	Three "?" cards. Team agreed they don't understand the ML requirements well enough.	Spike first (deferred)

What happened: The team estimated 4 stories (12 points) and identified 1 story to split and 1 that needs a spike before it can be estimated. Total estimation time: 18 minutes.

Facilitator tips

- **Anchor the scale early.** Before estimating new stories, pick 2-3 recently completed stories and assign them reference points. "That login page redesign was a 5. The copy change was a 1." Now the team has shared reference points.
- **Rotate who reads the story.** Don't let the PM always present. When an engineer reads the story, they notice gaps the PM missed.
- **Keep energy up.** Estimation sessions get tedious after 30 minutes. Estimate in batches of 8-10 stories, then take a break. Or estimate during grooming throughout the week, not all at once.
- **Celebrate the "?" card.** When someone plays it, they're saying "I don't know enough to guess." That takes courage. Thank them, capture what they need to know, and move on.
- **Track your calibration over time.** After a few sprints, compare estimates to actuals. Are 5-point stories consistently taking longer than expected? Your scale might need recalibration. This is normal and healthy.

How did it go?

After your estimation session, check:

- ☐ The team estimated using relative sizing, not hours or days
 - ☐ Everyone voted simultaneously (no anchoring)
 - ☐ Divergent estimates led to short discussions, not long debates
 - ☐ Stories estimated at 13+ points were flagged for splitting
 - ☐ "?" cards were used when clarity was missing (not just guessed through)
 - ☐ The session took less than 30 minutes for 8-10 stories
 - ☐ Nobody compared this team's velocity to another team's
-

Part of the [k8 Agent Toolkit](#). Download other formats at k8mak.com/resources.