

# TH;NKLOUDER

## The Scrum Master's Mini-Guide to Writing AI Prompts

How to write AI prompts that produce useful output: the four-part prompt structure, five common mistakes with rewrites, and six worked prompts for real Scrum situations.

Giora Morein • Certified Scrum Trainer (CST)

[thinklouder.com](https://thinklouder.com)

© 2026 ThinkLouder. All rights reserved.

---

FREE GUIDE

# The Scrum Master's Mini-Guide to Writing AI Prompts

How to write AI prompts that produce useful output: the four-part prompt structure, five common mistakes with rewrites, and six worked prompts for real Scrum situations.

Here's what I've noticed, watching people use AI tools in delivery work: the tool isn't the problem. The prompt is. Two people can open the same AI tool, ask about the same messy backlog item, and one gets a paragraph of generic advice while the other gets something they can actually bring into refinement. Same tool. Different prompt. That's the whole gap.

And so this guide isn't about which AI tool to use, because that changes every six months and honestly, it doesn't matter much. It's about how to ask. Whether you're a Scrum Master, a product owner, a business analyst, or a developer who just wants better answers out of the tool sitting in your browser tab, the skill is the same: you're having a conversation, and the quality of what comes back depends almost entirely on what you put in.

Think about it like briefing a new team member on their first day. If you say "go help the team," they'll wander around confused. If you say "here's our sprint goal, here's where we're stuck, here's what a good afternoon looks like," they can actually help. AI tools are the same. They're not mind readers. They're fast, capable, and completely dependent on you to tell them what they need to know.

I'll be honest with you, most people never get past "help me write a retro" or "summarize this." That's not a prompt, that's a wish. This guide gives you a simple structure for turning wishes into prompts that actually work, plus six situations pulled straight from the rhythm of Scrum, so you can see it in action before you need it.

## What makes an AI prompt actually work?

A good prompt has four parts. Not four rules to memorize, just four questions you're answering whether you realize it or not. Leave one out, and the AI has to guess, and when it guesses, you get generic output back. Here they are.

**Role.** Who should the AI be for this answer? Not because it changes what the AI "actually is," but because it changes the lens it answers through. Ask a prompt engineer, a retrospective coach, and a corporate memo writer the same question and you'll get three different answers. Same question, three completely different flavors. Example: "You're an experienced retrospective facilitator coaching someone running their first session."

**Context.** What does the AI need to know about your actual situation? This is the part people skip most, and it's the part that matters most. Team size, what just happened, what's already been tried, what's sensitive. Example: "My team of six just finished a sprint where three stories rolled over and there's some tension about it."

**Task.** What, specifically, do you want done? Not "help with the retro," but "suggest a retrospective format" or "draft three discussion questions." Specific verb, specific noun. Example: "Suggest a retrospective format suited to this mood, not our usual one."

**Format.** What shape should the answer take? A bulleted list, a short paragraph, a table, an email draft, a five-minute agenda. If you don't say, you'll get whatever shape the AI defaults to, which is usually longer and more generic than what you actually need. Example: "Give me a step-by-step agenda with rough timing for each step."

Put those four together and you've got a real prompt instead of a wish.

And here's the fifth thing, and it's not a fifth part, it's a practice: iterate on what you get back. The first answer is a draft, not a verdict. If it's close but the tone's off, say so. If it missed something about your team, add it and ask again. This is exactly the same instinct as inspect and adapt. You don't ship the first version of anything without looking at it first. Same deal here. Read what the AI gives you, tell it what's wrong, and let it take another pass. Two rounds of "actually, make it shorter" or "no, my team would never say it that way" will get you further than any perfectly worded first attempt.

## What are the most common prompt mistakes?

I see the same five mistakes over and over. And every one of them is fixable in about ten seconds once you know to look for it.

**Mistake one: the vague ask.** You type a handful of words, hit enter, and get back a couple of generic paragraphs that could apply to any team, anywhere. That's not the AI letting you down. That's the AI doing exactly what you asked for, which was, honestly, not much of anything.

**WEAK:**

"Help me with our sprint review."

**BETTER:**

"Draft a five-minute opening for sprint review that sets the stage for a demo of two completed features and one that's about 80% done, aimed at stakeholders who weren't in any of our daily scrums this sprint."

Look at the gap between those two. One tells the AI to improvise something generic. The other tells it exactly what room it's walking into.

**Mistake two: missing context.** You ask a sharp, specific question, and the answer still comes back a little off, because you left out the one detail that would have changed everything about how it should be written.

**WEAK:**

"Write a message about our blocked story."

**BETTER:**

"Write a message about our blocked story. The team's been waiting four days on API access from the platform team, and there are two more stories stacked up behind this one that can't start until it's unblocked."

Same task, same tone, but the second version knows what's actually riding on it, and that changes how it reads.

**Mistake three: no format.** You get back a wall of text when what you actually needed was three bullets you could read out loud in a two-minute stand-up.

**WEAK:**

"Summarize what changed in the backlog this week."

**BETTER:**

"Summarize what changed in the backlog this week. Three bullets max, plain language, no story point numbers, short enough to read out loud in under a minute."

**Mistake four: one-shot thinking.** You take the first answer and run with it, even when something about it feels a little off. Think about it: would you accept the first draft of a retrospective action item without checking whether the team actually agrees with it? Of course not. Same rule applies here.

**WEAK:**

accepting "Here's your working agreement" and pasting it straight into a wiki page without reading it twice.

**BETTER:**

"This is close, but agreement three assumes we're all in the same time zone. We're not, we've got two people three hours ahead of the rest of the team. Can you adjust it for that?"

**Mistake five: outsourcing judgement.** This is the one that sneaks up on people. It looks like you're just asking a plain question, but you're actually handing the AI a decision that isn't its decision to make.

**WEAK:**

"Should we drop this story from the sprint?"

**BETTER:**

"Draft three questions I can bring to the team to help us decide together whether to drop this story."

That last rewrite matters more than it looks. The AI can draft, it can suggest, it can organize your thinking into something useful. It shouldn't be making the call. That's still you, and more often than not, it's still your team, deciding together, in the room.

## Six prompts for real Scrum situations

Here are six spots where I've seen people reach for an AI tool, worked out as full prompts you can adapt. Swap in your own details wherever you see brackets.

Quick key before you dive in: the highlights in every strong prompt below show the four parts at work. **Role**, **context**, **task**, **format**. Once you see them separated, you can't unsee them, and that's the point.

# 1. Preparing a retrospective

**The situation.** Your team just closed out a rough sprint. Three stories rolled over, one developer pushed until 9pm the night before trying to save a fourth, and there's some unspoken tension in the room about it. Your usual "what went well, what didn't" format is going to fall flat with a room this frustrated, so you want a format that fits the mood, not the calendar.

## The weak prompt.

"Give me a retro format."

## The strong prompt.

"You're an experienced agile coach who's facilitated hundreds of retrospectives, including plenty of rough ones. My team of [number] just wrapped a sprint where [what happened, and the mood in the room, e.g. three stories rolled over, one person worked late trying to save a fourth, and people are frustrated with each other]. Suggest a retrospective format suited to this specific mood, not our usual go-to. For each step, tell me roughly how long it should take, what materials I need, and what I should say out loud to open it. Keep the whole retro to 45 minutes and leave room for at least one concrete thing the team decides to change. Ask me any additional questions you have."

**Pro tip.** Tell it the mood, not just the facts. "Frustrated" and "quietly discouraged" are both true after a rough sprint, but they call for two completely different formats, and the AI can't tell them apart unless you spell it out.



## 2. Triaging an impediment

**The situation.** Your team's been blocked for days: maybe it's access from another team, maybe it's a decision sitting on someone's desk, but work has genuinely stalled and everyone downstream feels it. Two other stories are already stacked up behind this one waiting to start, and if it doesn't move by the end of the week, the sprint goal is in real trouble. You need to escalate it, and you want the message to land as a clear ask, not a complaint that gets filed away and forgotten.

### The weak prompt.

"Help me write an email about this blocker."

### The strong prompt.

"You're an experienced agile coach helping me structure an escalation about a blocker. The message is going to a director who doesn't attend any of our team's meetings, so they're starting from zero. Here's the situation: [what's blocked, who's affected, how many days it's been blocked, what's already been tried, and what happens to the sprint if it's not resolved by a certain date]. Write a short email that states the business impact in plain terms, makes a clear and specific ask, names a deadline for a response, and stays polite even though this has been dragging on. Keep it under 150 words. Format it with a one-line subject, then the body, no more than three short paragraphs. Ask me any additional questions you have."

**Pro tip.** Lead with impact, not history: nobody escalating a blocker needs a play-by-play of everything that's already been tried, they need to know what's at risk right now and exactly what you need from them. Save the backstory for if they ask.

### 3. Translating team status for stakeholders

**The situation.** Sprint review is coming up, and half the room doesn't know a story point from a parking ticket. Two features shipped, one is stuck at code review and won't make the demo, and you need an update that tells stakeholders what actually matters to them, in language they'd use themselves. What you don't want is a status readout that buries the one thing they care about under delivery jargon they'll just nod along to and forget.

#### The weak prompt.

"Summarize what the team did this sprint."

#### The strong prompt.

"You're a communications-minded delivery coach helping me prepare a stakeholder-facing update for sprint review. My audience is [who's in the room, e.g. a VP and two customer representatives who don't attend our daily work]. Here's what the team completed and what didn't get finished, and why: [paste your list]. Translate this into three or four plain-language bullets that focus on outcomes and what it means for them, not process, story points, or engineering detail. If something slipped, say so plainly and say what happens next, don't bury it. No jargon, and nothing that assumes they already know how our team works. Ask me any additional questions you have."

**Pro tip.** If a stakeholder wouldn't say the word out loud in a hallway conversation, it probably doesn't belong in the update. That's a fast, reliable filter for jargon, and it works whether the news is good or bad.

## 4. Drafting refinement questions for a vague backlog item

**The situation.** A backlog item shows up for refinement and it's thin: a title and a single sentence, nothing more, maybe something like "let users export their data." You could hand the whole thing to the AI and ask it to write acceptance criteria, and it would happily invent a dozen of them, but that skips the conversation your team actually needs to have together. What you want instead is a sharp set of questions to bring into the room.

### The weak prompt.

"Write acceptance criteria for this story."

### The strong prompt.

"You're an experienced business analyst helping me prepare for backlog refinement. Here's a vague backlog item: '[paste the item as written]'. I don't want finished acceptance criteria, that's a conversation for the whole team to have together, not something to hand them pre-written. Instead, generate a list of clarifying questions I should bring into refinement, grouped by category: who the user actually is, what happens in edge cases, what could go wrong, and anything non-functional we might be missing, like performance or access. Ask me any additional questions you have."

**Pro tip.** Acceptance criteria frame the conversation, they don't replace it. If the AI hands you finished criteria anyway, that's your signal to redirect it and ask for questions instead, every time.

## 5. Facilitating a working-agreements session

**The situation.** You've either got a brand-new team that's never sat down to agree on how it wants to work, or an existing one that's quietly drifted from whatever ground rules it started with. Maybe people are talking over each other in stand-up, or half the team shows up late and nobody's said anything about it yet. Either way, you need to run a session that resets or establishes those agreements, and you'd rather it not feel like a lecture.

### The weak prompt.

"Help me run a working agreement meeting."

### The strong prompt.

"You're an experienced facilitator helping me design a working-agreements session for my team. Context: [team size, new or existing, remote/hybrid/co-located, any specific friction that's prompting this, e.g. people talking over each other in stand-up or showing up late without saying anything]. Design a 60-minute session agenda with time blocks, and for each block, give me two or three discussion prompts I can use to get the team talking instead of me lecturing at them. Include a way to capture what the team agrees to before the session ends. Ask me any additional questions you have."

**Pro tip.** Ask for prompts, not rules. A working agreement the team writes together in response to good questions sticks; one you hand them from a template doesn't.

## 6. Making sense of sprint metrics

**The situation.** You're looking at velocity or burndown data across the last few sprints, and something looks off. Maybe the numbers dipped and you don't know if it's a blip or an actual pattern, and you want to understand what's really going on before you bring anything to the team. The last thing you want is for this to land like a scorecard, because the moment metrics start to feel like a judgement, people stop being honest about them.

### The weak prompt.

"Analyze our velocity."

### The strong prompt.

"You're a data-literate agile coach helping me understand sprint metrics, not judge them. Here's our velocity for the last [number] sprints: [paste numbers], and some context: [anything relevant, like a team member out on leave, a holiday week, or a sprint where scope changed midway]. Identify two or three patterns worth discussing with the team. Frame each one as a question I could ask the team, not a conclusion I've already reached, and avoid language that sounds like I'm building a case against anyone. This is for a conversation, not a performance review. Ask me any additional questions you have."

**Pro tip.** Metrics point at questions, not verdicts. If a pattern looks bad, the AI's job is to help you ask about it, not to explain why the team fell short.

## The bonus tip: let the AI ask you questions

Here's a small thing that changes a lot: end your prompt with "Ask me any additional questions you have."

Think about what that flips. Without it, the AI fills every gap in your prompt with a guess, and you don't find out what it guessed until you're staring at an answer that's almost right but not quite. With it, the AI tells you what it's missing before it writes a single word, and you get to fill the gap yourself instead of hoping it guessed correctly.

You'll notice every strong prompt in this guide ends with that exact line. That's not an accident, and it's not decoration. It's doing real work in every one of them. Use it on prompts you write yourself too, not just the ones here. It costs you nothing and it catches the context you forgot to include, which, let's be honest, is most of us, most of the time.

## Where should you be careful?

Three things worth keeping in mind as you use any of this.

**Never paste confidential company data into an AI tool.** Salary numbers, unreleased roadmap details, anything with a customer's name attached, anything your company would ask you to sign an NDA about. Strip it out or use placeholders before it goes anywhere near a prompt. When in doubt, leave it out.

**AI drafts, you decide.** Everything in this guide produces a starting point: a draft agenda, a draft email, a set of questions. None of it is a decision. The scenario about metrics and the scenario about backlog refinement both make this explicit for a reason: the AI can organize your thinking, but the call is still yours, and often it's your team's call to make together, not yours alone.

**The conversation with the team is the job.** A well-written prompt gets you into the room better prepared. It doesn't replace the room. If you find yourself using AI output as a substitute for the retrospective, the refinement session, or the hard conversation instead of preparation for it, that's worth noticing.

## The cheat sheet

Four parts, one practice, one line worth memorizing.

- **Role:** who should the AI be for this answer?
- **Context:** what does it need to know about your situation?
- **Task:** what specifically do you want it to do?
- **Format:** what shape should the answer take?
- **Iterate:** the first answer is a draft. Read it, correct it, ask again.
- **The line to end every prompt with:** "Ask me any additional questions you have."

Print it, screenshot it, stick it next to your monitor. Next time you catch yourself typing "help me with," stop, and answer those four questions first.

Get the AI for Delivery Newsletter: [thinklouder.com/ai-prompt-writing-for-scrum-masters/#newsletter](https://thinklouder.com/ai-prompt-writing-for-scrum-masters/#newsletter)