Prompt Engineering Best Practices Commentary and Discussion Questions

Kevin Crook

KevinCrook.com

Start Simple with Interactive Chat

People freeze at the blank box. They believe there is a right way to ask and that they do not know it, so they type something vague, get something vague back, and conclude that AI is overrated.

So the first best practice is also the most freeing one. Do not try to write the perfect prompt on the first try. Nobody does. Trial and error is normal and expected, and it is not a sign that you are doing this wrong. It is the method.

Begin with a basic prompt, then refine it step by step. Ask for the thing. Look at what comes back. Then say what was missing.

That last part is where people go wrong. You are not starting over each time. Each prompt inherits everything before it, so six short, ordinary sentences can build something genuinely elaborate.

Let me show you.

Live Demo

Prompt 1: The basic first attempt

> Plan a haunted house for a high school food bank drive. The haunted house will be free to community kids as a public service, but optional food bank donations will be accepted.

Prompt 2: Add the two-gym, two-experience structure

> We have two gyms available, one large and one small. Can you rework this so the small gym houses a non-scary, friendly haunted house designed for younger kids, and the

large gym houses a scarier haunted house for older kids and teens, but still family-friendly enough for a community event, not intense or graphic?

Prompt 3: Add the budget

> Our school has about 2,000 students, and a school-wide donation drive raised about $2,500 toward decorations. Can you rework the plan around that budget, splitting it reasonably between the two gyms?

Prompt 4: Add a flow and pacing element

> Can you build in a way to keep groups moving every few minutes in both gyms so we don't get lines backed up outside, without making anyone feel rushed?

Prompt 5: Add a story for each room

> For each room in both haunted houses, can you add a short backstory or scenario our student volunteers could act out or reference: spooky but not scary for the small gym, and fun-scary but family-friendly for the large gym?

Prompt 6: Tighten the format

> Can you reformat this as a simple numbered walkthrough for each gym, with room name, scare concept, props needed, and estimated cost, so our volunteer team can use it as a planning checklist?

The Secret Sauce of Prompt Engineering

Everything up to now has been groundwork. This slide is the reason the workshop exists.

Three techniques, used together. Meta-prompting, plus Role, plus COSTAR. I call it the secret sauce of prompt engineering, and the triple power is in the combination rather than in any single piece.

This is not my invention and it is not a theory. Major prompt engineering contest winners and top finishers use this technique. When there was real money and a real ranking on the line, this is what came out on top.

It is the most important takeaway from this workshop. If you remember nothing else from today, remember those three words together.

The next three slides take them one at a time. Meta-prompting first, because it does most of the work, and then Role and COSTAR, which are what meta-prompting produces.

Meta-Prompting

Here is the move, and it is close to a magic trick.

Stop trying to write a great prompt. Ask the AI to write it for you.

Think of AI as your personal expert prompt engineer. It has read essentially everything ever written on the subject, including the research and the contest write-ups. AI can do prompt engineering better than most humans can. I would go further and say maybe all of us, and I teach this for a living.

So describe what you want in plain, messy, ordinary language. Then add the one instruction that changes everything: instead of answering this yourself, write the expert prompt I should use.

Now the part people get backwards. Use two chats.

Chat one is where you ask the AI to help you create and refine prompts. Chat two is where you run the finished prompts.

Why separate them? Because in chat one the AI has already seen your messy description and everything around it. If it answers there, it is answering with all of that in front of it, and you learn nothing about whether the prompt itself is any good. A fresh chat tests the prompt the way it will actually be used.

Watch.

Live Demo

Prompt 1 (run in Chat 1):

> I want to plan a haunted house for a high school food bank drive, using two gyms: a small gym with a non-scary, friendly haunted house for younger kids, and a large gym with a scarier but family-friendly haunted house for older kids and teens. The event is free to the community as a public service, with optional food bank donations. Our school has about 2,000 students, and a school-wide donation drive raised about $2,500 toward decorations, to be split reasonably between the two gyms.
>
> Instead of answering this yourself, write an expert prompt I should use to get the best possible result from an AI chatbot for this request.

Prompt 2 (run in Chat 2):

> [copy and paste the output from prompt 1 to chat 2 and run it]

Role + COSTAR

Meta-prompting works better when you tell the AI what shape to build. Role plus COSTAR is that shape, and it is seven pieces.

Role

Assign the AI a specific role or expertise. You are an experienced travel agent. That single line changes its vocabulary, its assumptions, and what it thinks is worth mentioning.

Then COSTAR, which is an acronym covering the six that follow:

Context

Background information the AI needs. Everything that is obvious to you and impossible for it to know.

Objective

Exactly what you want done. Not the topic, the task.

Style

The writing style you want. Formal report, casual, technical.

Tone

The attitude of the response. Friendly, professional, persuasive. Style is how it is written. Tone is how it feels.

Audience

Who the response is for. Beginners, experts, kids, your boss. That one field does more work than any of the others.

Response Format

How to deliver the answer. Bullet list, table, paragraph, long form, with examples.

Meta-Prompting

Seven fields, and here is the honest part. You do not have to fill them in yourself. You ask the AI to fill them in for you, using meta-prompting, and then you correct whatever it got wrong.

That is the triple power. You supply the mess, the AI supplies the structure, and you keep the final say.

Live Demo

Prompt 1(run in Chat 1):

> I want to plan a haunted house for a high school food bank drive, using two gyms: a small gym with a non-scary, friendly haunted house for younger kids, and a large gym with a scarier but family-friendly haunted house for older kids and teens. The event is free to the community as a public service, with optional food bank donations. Our school has about 2,000 students, and a school-wide donation drive raised about $2,500 toward decorations, to be split reasonably between the two gyms.
>
> Instead of answering this yourself, write the ideal prompt I should use, structured using the COSTAR framework with these exact labeled sections:
>
> Role:
> Context:
> Objective:
> Style:
> Tone:
> Audience:
> Response:
>
> Fill in each section based on my request above, adding reasonable specific details where helpful.

Prompt 2 (run in Chat 2):

> [copy and paste the output from prompt 1 to chat 2 and run it]

In Context Learning

Back in the basics section we said the answer to a training cutoff is to hand the model the material. This is the name for that.

AIs learn from whatever you put in the context window, with no retraining needed. Paste a document in and the AI knows it, right now, in this conversation, as well as if it had been trained on it.

File uploads and Projects are the everyday version. Your documents become knowledge the AI can use. Your company handbook, your class notes, a contract you do not understand.

Two things worth knowing. It is temporary and local to that conversation, which is a privacy feature as much as a limitation. And what you provide beats what it was trained on, so it will follow your document over its own general knowledge.

Now the move that is easy to miss. Meta-prompting works here too. Ask the AI to help generate the context material itself: summaries, examples, background documents. You can have it write the briefing that you then feed back to it in a fresh chat.

That sounds circular. It is not. You are using it to prepare, and then using the preparation.

Here is the difference it makes.

Live Demo

You will need a short story the AI has never been trained on, plus a list of questions about it, and a list of answers to compare against.

The workshop uses "The Vampire at Putnam County High," about Silas Vane, the vampire living in the basement of Putnam County High School:

> The Vampire at Putnam County High (short story)
>
> Questions
>
> Answers

Prompt 1: Ask about a story it has never seen

> Answer the following questions about the short story "Silas Vane" (the vampire who lives in the basement of Putnam County High School). Answer each question as best you can based on what you know. If you don't know, make your best guess rather than saying you don't know. For each answer, tell me how you arrived at it.
>
> Questions: [trying asking a few questions 1 by 1, then try asking all of them at once]

Prompt 2: Give the AI the story in context, then ask again

> Below is a short story. Read it carefully, then answer the list of questions that follows using only information from the story.
>
> Story: [paste the full vampire story here or upload a file containing it]
>
> Questions: [paste the questions list here or upload a file containing them]

Manage Context Rot and Task Rot

You already know what context rot and task rot are. Here is what to do about them.

First, do not wait for trouble. Periodically ask the LLM to summarize the conversation so far. This keeps both of you focused, and that is not a figure of speech. It pulls the important material back to the front of the window, and it shows you whether the AI still understands the job.

Second, know when to quit. At the first signs of context rot or task rot, cut your losses. Ask for a summary, start a fresh new chat, and paste it in.

People resist this. There is a feeling that a long conversation is an asset and that starting over throws it away. The opposite is true. Most of a long chat is noise by the end, and the summary is the part worth keeping.

Here is how it looks.

Live Demo

Prompt 1: In the current chat, ask for the summary

> Before we continue, please summarize this conversation so far: what we've been working on, the key decisions we've made, anything I've asked you to remember or apply going forward, and where we currently stand. Write it so I could paste it into a brand new chat and you'd be able to pick up exactly where we left off.

Prompt 2: In a brand new chat, restart from the summary

> Here's a summary from a previous conversation. Please read it, confirm you understand where things stand, and let's continue from here:
>
> [paste the summary]

Handle Lost in the Middle Issues

Same situation, different problem. This one is for large documents or long contexts, where the middle quietly gets ignored.

Three steps. Break them into smaller chunks. Process each piece separately. Then ask the LLM to combine or summarize the results.

The important word is separately. Each chunk gets its own fresh chat, so nothing is ever in the middle of anything. Every piece is the only thing in the window, which means every piece gets full attention.

Then the combining pass works on the summaries rather than on the original, and summaries are short enough that there is no middle left to lose anything in.

It feels like more work than pasting the whole document in and hoping. It is more work. It is also the difference between a summary of the document and a summary of its first and last few pages.

Let’s look at some example prompts we can use.

Example Prompts

Prompt 1: In a separate chat for each chunk

> Here is a chunk of a longer document. Please summarize the key points from this chunk only, in a clear and concise way. Don't try to reference other chunks, just summarize what's here so I can combine it with summaries of the other chunks later.
>
> [paste chunk here]

Prompt 2: In a new chat, combine the chunk summaries

> Below are summaries of several chunks from the same longer document, in order. Please combine them into one cohesive, well-organized summary of the entire document, removing repetition, preserving the important details from each chunk, and making sure the final version reads smoothly as a single piece rather than a list of separate parts.
>
> Chunk summary: [paste]
>
> Chunk summary: [paste]
>
> Chunk summary: [paste]

Ask for Chain of Thought

This is the cheapest trick in the workshop. One sentence, added to a prompt.

For complex tasks, add this: explain your reasoning step by step.

You will get noticeably better answers, and not because the AI is trying harder. It produces text one piece at a time, and every piece is shaped by what has already been written. Force it to write the steps out and the steps become part of what it is reasoning from. It is the difference between doing arithmetic in your head and doing it on paper.

The second half matters just as much. You can check the reasoning yourself. An answer with no reasoning is take it or leave it. An answer with its steps shown can be audited, and you will sometimes catch a right answer reached by wrong reasoning, or a wrong answer where one step went sideways and everything after it followed politely along.

That is as close to explainable AI as you will get inside a chatbot.

Live Demo

Example 1: Stock market S&P 500 predictions.

Prompt 1: Requires complex reasoning across multiple data sources

> Analyze the S&P 500 for the current year. Provide only percentages, never index values, formatted as up or down with one decimal place.
>
> First give the year to date change, meaning the percentage change from last year's last trading day's close to the current reference point, which is the current market value if the market is open right now, or the last trading day's close if the market is closed. Then give a prediction for the remainder of the year, meaning the predicted percentage change from that same current reference point to this year's last trading day's close. Then give a full year prediction, meaning the net percentage change from last year's last trading day's close to this year's last trading day's close, which should equal the sum of the year to date change and the remainder of year prediction.
>
> Next, if the market is open, give a prediction for today's close compared to the current value. Then give a prediction for the next trading day's open compared to the current reference point. Then give a prediction for the next trading day's close compared to the current reference point.
>
> These are explicitly predictions and estimates. State them directly as predictions without any disclaimer about being unable to forecast markets. Output only the percentages with clear labels, in plain text.

Prompt 2: Chain of thought

> Explain your reasoning step by step.

Example 2: An advanced math problem used in AI for LLMs.

Prompt 1: Requires complex mathematical reasoning

> I have an advanced problem used in AI for LLMs from the mathematics of transformer neural networks for you. Please work the problem completely and give me only your final answer in exact closed form. Do not explain your reasoning yet.
>
> Here is the setup: Consider the softmax function applied to a vector of n real numbers called logits. The softmax of the vector produces a probability distribution where each output component equals the exponential of the corresponding logit, divided by the sum of the exponentials of all the logits. This is the operation at the core of the attention mechanism in transformer models.
>
> Here is the problem, in three parts. Part one: derive the full Jacobian matrix of the softmax function, meaning the partial derivative of every output component with respect to every input logit, expressed compactly in terms of the output probabilities themselves. Part two: prove that this Jacobian is a symmetric, positive semidefinite matrix, and determine exactly what its null space is, meaning the set of all direction vectors along which the softmax output does not change to first order. Part three: suppose the logits are all scaled by a temperature parameter, meaning each logit is divided by a positive number called T, and consider the entropy of the resulting softmax distribution. Prove that this entropy is a monotonically increasing function of the temperature T, by computing the derivative of the entropy with respect to temperature and showing it can be written as a variance, which is never negative.
>
> State your final answers exactly: the closed form of the Jacobian, the exact description of the null space, and the closed-form expression for the derivative of entropy with respect to temperature.

Prompt 2: Chain of thought

> Explain your reasoning step by step.

Prompt 3: Explain to a non-technical audience

> Explain this in plain simple english to a non-technical audience.

Try Different LLM Versions and Vendors

LLM vendors offer multiple versions. Fast ones, deep ones, beta ones. The fast version is usually the default because it is cheap to run, and for a hard question it is the wrong choice.

And there are multiple vendors to choose from. ChatGPT, Claude, Gemini, Grok, with more arriving.

So if you are stuck or unsatisfied, remember that the same prompt can get very different results elsewhere. Not slightly different. Sometimes one system produces something genuinely useful where another produced nothing at all, from identical text.

The habit worth building is to treat a bad answer as information about the pairing rather than about your question. Before you rewrite the prompt for the fourth time, paste it somewhere else.

The free tiers make that nearly costless, and it takes about ten seconds.

Build or Use a Prompt Library

You are going to write good prompts and then lose them. Everybody does. You will remember getting a great result and have no idea how you asked for it.

So save your best prompts in a document as you go, for easy reuse and adaptation. Not afterward. As you go, because afterward never arrives.

What you are really building is a personal toolkit. Prompts are reusable in a way that answers are not. An answer helps you once. The prompt that produced it helps every time that situation comes around again.

You can also search for public prompt libraries for great starting points. Plenty of people publish theirs. Take one, adapt it to your situation, and keep the version that worked.

One habit makes the whole thing worth having. Write a line next to each prompt saying what it was for. A prompt with no note attached is nearly as lost as no prompt at all.

Share Chat Links

One last one, small and underused.

Share useful chats using the LLM's share link. Most chatbots have one built in, usually near the top of the conversation.

What makes that better than copying the answer out is that it carries the whole thing. It is great for showing coworkers, family, or friends exactly how you got a result, prompts and all.

That matters more than it sounds. Sending somebody an answer teaches them nothing. Sending them the conversation teaches them how you got there, and that is the part they can actually reuse.

It is also the fastest way to teach prompt engineering to somebody who was not in this room.

One caution before you send anything. A share link carries the entire conversation, including whatever you typed earlier and forgot about. Read it through before you hand it to your boss.

Discussion Questions

The exercises below come in three sizes, and every section has one of each. The time estimates describe a room rather than a clock, and they will move with the size and the mood of your group.

Use them however they fit your schedule. A ten to fifteen minute exercise can be homework, or it can be the centerpiece of a class session. A five to seven minute exercise works in small groups or with the whole room together. Nothing here is tied to a particular format.

Every exercise ends with something produced. Everything needs nothing more than paper and a pen unless the exercise says otherwise, though several become better if devices are available, and those are marked.

Start Simple with Interactive Chat

1. **(estimated 1 to 2 minutes)** Name a time an AI gave you a useless answer. What was missing from the question rather than from the answer?
2. **(estimated 5 to 7 minutes)** Take a deliberately vague request and write six prompts that build on each other, each one adding exactly one thing. Report which single addition you think would change the result the most, and defend it.
3. **(estimated 10 to 15 minutes)** Pick a real task. First, write the one perfect prompt you would attempt if you only had one shot. Then write the six-step interactive version. Trade both with another group and have them judge which would produce better results. Finish with a paragraph on which was easier to write, and whether easier and better were the same answer.

The Secret Sauce of Prompt Engineering

4. **(estimated 1 to 2 minutes)** Three techniques combined. Before we go further, which do you predict does the most work?
5. **(estimated 5 to 7 minutes)** Write what you expect each of the three to contribute. Then rank them by importance. Report your ranking and defend whichever one you put last, since that is the one you are claiming matters least.
6. **(estimated 10 to 15 minutes)** Find a task where this combination would be overkill or would actively make things worse. Describe it. Then write the simplest prompt that would do the job properly. Finish with a paragraph on how you tell in advance whether a task deserves structure.

Meta-Prompting

7. **(estimated 1 to 2 minutes)** Would you rather be handed a good answer or a good question? Say why in one sentence.
8. **(estimated 5 to 7 minutes)** Write a messy, plain-language description of something you actually want. Then write the single sentence that converts it into a meta-prompt. Report your sentence and compare wordings with another group.
9. **(estimated 10 to 15 minutes)** Write your messy description, then write your own best attempt at the expert prompt it should produce. Trade with another group and have them write what they would expect an AI to produce from your description. Compare all three. Finish with a paragraph on what the other group included that you left out. If devices are available, run it and add the AI's version to the comparison.

Role + COSTAR

10. **(estimated 1 to 2 minutes)** Seven fields. Name the one you think most people forget.
11. **(estimated 5 to 7 minutes)** Take a request and fill in all seven fields for it. Then remove one field and describe specifically how the answer would change. Report the field whose removal did the most damage.
12. **(estimated 10 to 15 minutes)** Write the same request three times, changing only the Audience field: once for a child, once for an expert, once for someone who has to make a decision today. Predict each response in a few sentences. Finish with a paragraph on how much of the output is decided by that one field alone.

In Context Learning

13. **(estimated 1 to 2 minutes)** Name a document you would want an AI to know that it could not possibly have been trained on.
14. **(estimated 5 to 7 minutes)** You have a document the AI has never seen. Write five questions it could only answer with the document in hand, and five it could bluff its way through. Report the pattern that separates the two lists.
15. **(estimated 10 to 15 minutes)** Write a one-page briefing that would let an AI answer questions about something you know well. Then trade, and have another group write three questions your briefing cannot answer. Finish with a paragraph on what you left out, and on whether you left it out because you forgot or because you assumed.

Manage Context Rot and Task Rot

16. **(estimated 1 to 2 minutes)** Name the first sign that would tell you a long conversation has drifted off track.

17. **(estimated 5 to 7 minutes)** Write the summary request you would use. Then list everything a good summary has to contain for the work to restart cleanly. Report your must-have list and see what other groups included that you did not.
18. **(estimated 10 to 15 minutes)** Take a long project you know well and write the summary you would carry into a fresh chat. Trade with another group and have them attempt to describe the next step using only your summary. Finish with a paragraph on what broke, and on why you thought it was safe to leave out.

Handle Lost in the Middle Issues

19. **(estimated 1 to 2 minutes)** You have a hundred-page report. Where would you cut it into chunks, and what is your rule?
20. **(estimated 5 to 7 minutes)** Design a full chunking plan for a specific document: where the boundaries go, what you ask about each chunk, and how you combine the results. Report your boundary rule and defend it against a group that chose differently.
21. **(estimated 10 to 15 minutes)** Take something real and long. Chunk it, summarize each chunk yourself, then combine your summaries into one. Then summarize the whole thing in a single pass without chunking. Compare the two. Finish with a paragraph on what the chunked version caught that the single pass missed, and on what it cost you.

Ask for Chain of Thought

22. **(estimated 1 to 2 minutes)** Have you ever reached a right answer through wrong reasoning? What happened?
23. **(estimated 5 to 7 minutes)** Take a multi-step problem and write out the steps to solve it. Then go back and insert one plausible error partway through, carrying it forward so everything after it stays internally consistent. Trade and see whether another group can find it. Report where yours hid.
24. **(estimated 10 to 15 minutes)** Take an answer that shows its reasoning, either from an AI or one your instructor provides. Audit it one step at a time and mark every step that is asserted rather than supported. Finish with a paragraph on whether the reasoning shown is necessarily the reasoning used, and on how you could possibly tell.

Try Different LLM Versions and Vendors

25. **(estimated 1 to 2 minutes)** How many different AI chatbots have you actually tried? Count hands at one, two, three, and more.
26. **(estimated 5 to 7 minutes)** List the reasons the same prompt might produce different answers on different systems. Then rank those reasons by how much each should affect your trust in any single answer. Report your top reason.
27. **(estimated 10 to 15 minutes)** Design a fair test. Write one prompt you would use to compare vendors, chosen so that real differences would actually show up, and write down exactly what you would measure. Then trade and critique another group's test.

Finish with a paragraph on whether better is even one thing, or several things that disagree.

Build or Use a Prompt Library

28. **(estimated 1 to 2 minutes)** Name a prompt you would want to keep forever.
29. **(estimated 5 to 7 minutes)** Design how your library would be organized: what the categories are, and what each entry stores besides the prompt itself. Report your categories and see whether anyone found one you missed.
30. **(estimated 10 to 15 minutes)** Write five prompts you would genuinely reuse, in full, with blanks where the details change. Trade with another group and have somebody try to use one of yours for their own situation. Finish with a paragraph on what they had to fix, because that is what your prompt was assuming without saying.

Share Chat Links

31. **(estimated 1 to 2 minutes)** Name one person you would send a chat to, and say what they would get out of it.
32. **(estimated 5 to 7 minutes)** Compare sending somebody the answer against sending them the whole conversation. List what each one teaches the person receiving it. Report the biggest gap between the two lists.
33. **(estimated 10 to 15 minutes)** Pick something you know how to get from an AI that somebody else does not. Write out the conversation you would want them to see, prompts included, as though you were sharing the link. Then trade and have them describe what they learned. Finish with a paragraph on what the conversation taught that a finished answer never could.

References

Coming soon

Copyright and Permissions

You are also welcome to adapt my materials to fit your own students. If you do, please include the notice *Adapted from content by Kevin Crook (KevinCrook.com)* prominently under the title of the document, or under the title of the section it belongs to, and at the bottom of any slide or group of slides adapted from the original.