

# DON'T REPLACE PROBLEM SOLVING WITH AI

By Jamie Flinchbaugh

Adopted from an article published by The LeanMag



JFlinch

# DON'T REPLACE PROBLEM SOLVING WITH AI

by Jamie Flinchbaugh



Often people are looking at AI as a faster way to get to an outcome, such as a solution to a problem. There are several challenges with this. First, problem solving isn't one skill but an assembly of many skills and steps, and asking AI to replace it in one step will make it a victim of the same mistakes humans make when jumping to solutions. These capabilities, such as shaping the problem and deeply understanding cause and effect, I explore in [People Solve Problems](#). Second, if problem solving is a learning activity, you can't get any learning when it all happens within a black box. Third, it fundamentally isn't using logic or creativity, and you are severely limiting

yourself to the most median of solutions and thinking within the process. That doesn't mean AI can't be helpful to your problem solving process.

It means that Artificial Intelligence should be treated like a team member on a problem solving team rather than a replacement of your ownership of the problem. Think of all the questions you might ask a team while working on problem solving. Those are the same questions you can ask of AI. The term that I've found really useful when thinking about this type of AI use is co-intelligence (borrowed from the [book by Ethan Mollick](#)), as it is being

used in cooperation with your own intelligence.

Almost every problem solving methodology highly values showing your work in carefully laid out steps. While I think we are sometimes too dogmatic about how we follow those steps, there is tremendous value in breaking it down and showing your work. This is because problem solving is an act of learning and discovery, and both the key insights and the mistakes are seen in the path followed as opposed to just the outcomes. Instead of using AI to jump to solutions, which is the mistake we're trying to avoid as humans, we should use it as a co-intelligence along the steps of problem solving.

As a demonstration of this, let's imagine you want to reduce rework in your product development process. If I ask [Claude](#) (or ChatGPT, or whatever tool you prefer) "how should I reduce rework in my product development process?" I will get some very broad and fundamentally obvious solutions. In my quick test, recognizing that the prompt could be designed much better, I received solutions such as improving requirements gathering, enhancing communication, implementing iterative development, conducting thorough testing, and utilizing prototyping. All of these are sound suggestions that we should probably already be trying to do, and it comes back written clearly and authoritatively, often giving the user a false confidence that they have found the answer. But these broad and generic solutions are simply good ideas that may be completely disconnected from your problem.

Instead, we can use AI as a co-intelligence along steps of problem solving to sharpen and refine our own work. Let's start with the vitally important problem statement. I might start with describing the situation building up to a problem statement:

*"We have rework loops in our product development process that can add full design iterations, and almost always one iteration, to our full development time. This is causing us to miss deadlines and increases our spending. It also increases frustration within the team. What questions should I answer before I define a problem statement?"*

You may notice that this turns your AI co-intelligence into your problem solving coach. It may not be as insightful as a problem solving coach with expertise in product development, but that isn't always available. Here are some of the questions it asked me:

1. What specific stages of the PD process are most affected by the rework loops?
2. How frequently do these rework loops occur?
3. What are the main causes of these rework loops? (you may notice this is the absolute wrong time to ask this question, which illustrates my thesis that AI can't replace your ownership of the path forward)
4. How much additional time and cost do these rework loops typically add to a project."
5. Which team members or departments are most impacted by the rework?

Followed by 7 more questions. Some questions are more useful than others but

“Instead of using AI to jump to solutions, (...) we should use it as a co-intelligence along the steps of problem solving.

they are there. Like any good coach, asking these questions can help ensure that we don't get hooked into too narrow a focus too early.

Next I can answer those questions which give me lots of context, and it sets up my thinking to generate a good problem statement. The problem statement is critically important as it frames all the work that will follow. As John Dewey declared, "a problem well defined is a problem half solved." You should take ownership over what good problem statements look like. If you have company standards for problem solving, you can take the criteria for a good problem statement and make it available to people as a cut-and-paste prompt. Here's an example:

*"Take all of the information I've provided above and generate 7 different problem statement suggestions. A problem statement should be clear and concise. It should define a gap, with numbers if possible, between the current performance and the standard or expected performance. It should avoid any assumptions about the cause or suggestions of solutions."*

The AI tool will now provide us several problem statements to choose from, or more likely, to tweak and refine to the problem statement that you finally decide on.

While this might seem like extra work rather than a faster path to just write the problem statement, you are correct. That's actually the point.

Problem solving needs to be deliberate, rigorous, and iterative. Acting as a co-intelligence rather than a human replacement AI can be more like a team member, a coach, or a thought-partner. □