

When AI Handles the Work: How Program Managers Stay Valuable ¹

Ankur Gupta

Senior Staff Technical Program Manager, LinkedIn

Introduction: AI raises the bar for program management

For most of my career, program managers have been easy to recognize. We were the people carrying the plan, taking notes, chasing updates, maintaining the risk register, and turning a week of conversations into a Friday status report.

All of that work still matters. Programs still need accurate plans, clear records, timely updates, and disciplined follow-through. The people doing it were solving real coordination problems, often through substantial manual effort. What is changing is not the importance of the work, but how much time and effort it requires.

AI is making that kind of coordination dramatically cheaper. It can summarize a meeting, draft an update, compare plans, organize risks, and turn scattered information into a readable brief. Work that once occupied an afternoon can now take minutes.

This is usually framed as a threat to the program manager. I see it differently. AI is forcing us to separate the work that kept us busy from the work that made us valuable.

Once the administrative layer becomes easier, the question is not whether the program manager still has a job. The question is what the job was really for.

AI makes coordination easier, not optional

Nobody on my programs reports to me. I cannot offer a raise, grant a promotion, or rely on a reporting line to make ten teams move together. Yet I may still be responsible for getting a vice president to make a decision, carrying an architect's direction into rooms they will never attend, and keeping several organizations pointed at one outcome.

¹ How to cite this paper: Gupta, A. (2026). When AI Handles the Work: How Program Managers Stay Valuable; *PM World Journal*, Vol. XV, Issue IX, September.

That works because a program manager quietly holds a small part of many other people's responsibilities.

The executive has the authority to align teams but not the time to sit in every working session. The architect owns the technical direction but cannot explain the reasoning in every adjacent meeting. Each engineering manager owns a team's delivery but may not see the complete chain of dependencies around it.

The program manager connects those incomplete views. We carry context, detect when two teams believe different versions of the same decision, surface resistance that never appears in a status update, and create the conversation that allows work to continue.

This is different from moving information. Moving information is producing the minutes. Judgment is noticing that the polite agreement recorded in those minutes is not a real commitment.

AI is already good at the first part. It is not accountable for the second.

Move from tracking decisions to shaping them

I once led a product launch involving six teams and a hard deadline. Two weeks before launch, the work stopped because two principal engineers could not agree on an architectural direction. Both designs were credible. Neither engineer could make the final call alone, and the people with the necessary authority were not in the room.

The familiar response would have been to send a follow-up email, mark the item yellow, and report that the decision was still in progress. That would have described the blockage accurately. It would not have changed anything.

Instead, I worked backward from the decision. What exactly had to be decided? Who was accountable for the call? What were the two viable options? Which criteria mattered most for this launch: reliability, latency, cost, delivery time, or future flexibility? When would delay begin to damage the launch? What would happen if we made no decision at all?

I brought the engineers and the accountable directors into a thirty-minute meeting with a decision required by the end of the day. My role was not to choose the architecture. I did not own the technical answer. I owned the decision-making system: the participants, the options, the criteria, and the clock.

This distinction is becoming more important in the AI era. AI can draft a decision document, summarize the arguments, and identify areas of disagreement. But a real

decision also depends on context that may never appear in the documents: who can accept the risk, which tradeoff the organization will tolerate, whether the disagreement is technical or personal, and when continued analysis becomes more costly than choosing.

A program manager who only reports that a decision is late is narrating the program. A program manager who creates the conditions for the decision is leading it.

Domain shallowness is no longer an option

Strategic judgment does not require the program manager to become the deepest expert in every room. That would be unrealistic, and it would blur the responsibilities of the people who actually own the domain. The goal is to understand enough to identify the question that changes the decision.

Think about removing a wall during a home renovation. You do not need to become a structural engineer or calculate the size of a replacement beam. But you do need to know enough to ask, "Is this wall load-bearing?" Out of dozens of questions, that one determines what can safely happen next.

Technical and domain depth work the same way. A program manager does not need to write the code, design the database, or build the financial model. But the person leading the program should understand what a decision costs in reliability, technical debt, team velocity, money, or future flexibility.

"Can it be done?" is rarely enough. Better questions are: What are we giving up? Is this decision reversible? What has to be true for this plan to work? When does today's shortcut become tomorrow's constraint?

AI has made it easier than ever to prepare for these conversations. It can explain an unfamiliar architecture, compare alternatives, translate terminology, and keep answering basic questions without impatience. That removes the excuse for remaining shallow on the domain.

What AI cannot determine is which question is load-bearing for this particular program. The answer depends on the deadline, the customers, the history between teams, the organization's risk tolerance, and constraints that may not be written down. That is where understanding becomes judgment.

Bring product thinking and business judgment into program leadership

Program managers are trained to care about whether the team reaches the destination on time. Strategic leaders also ask whether it is the right destination.

My team once spent a full quarter building a monitoring dashboard. Stakeholders had requested it. We included the charts, filters, and views they asked for, and the project shipped on schedule.

A month later, I checked the usage. The dashboard had been viewed nine times.

When we finally observed how the intended users actually worked, the problem became obvious. They did not need another dashboard to visit. They needed the one alert that required action to appear inside the tool they already used. We had delivered the requested output and missed the actual need.

This is the difference between managing delivery and exercising product judgment. A schedule can tell us when the dashboard will ship. It cannot tell us whether anyone will use it.

Business judgment adds another layer. Sometimes there are several worthwhile destinations and only enough people, time, or money to pursue one. The decision cannot be made by asking which request arrived first or which stakeholder argued most loudly. It requires comparing customer impact, cost of delay, risk reduction, and strategic value.

The facts may be the same, but the framing changes the conversation. “We are two weeks behind” communicates status. “We may miss the Q3 launch window and lose \$X million in revenue unless we choose Option A or Option B” creates a leadership decision.

AI can help organize the evidence and test the argument. It cannot be accountable for deciding which outcome is worth the investment. Program managers who want to operate strategically must connect execution with the reason the work exists.

Build the trust, relationships, and influence AI cannot replace

Good judgment has limited value if nobody trusts the person offering it.

Program managers rarely have enough formal authority to force alignment across every team involved in a complex initiative. We operate through credibility, relationships, and consistency. Those sound-like soft qualities until a program reaches a hard moment.

When two teams disagree, the visible argument is often not the complete problem. One side may be protecting an operational burden that the other side does not see. A stakeholder may resist a proposal because a previous commitment was not honored. A team may say yes in the meeting while quietly believing the plan is impossible.

People reveal those concerns to someone they trust to represent them accurately.

That trust is not built during the escalation. It accumulates through small choices: following through when you said you would, giving credit to the people who did the work, protecting a confidence, carrying another team's position accurately into an executive room, and addressing conflict directly instead of turning it into gossip.

AI can help prepare for a difficult conversation. I can use it to challenge my framing, anticipate objections, or find language that lowers the temperature. But it cannot establish the history that makes another person willing to be honest with me. It cannot spend political capital, repair a damaged relationship, or take responsibility when a call goes wrong.

That is why organizational influence becomes more valuable as information becomes easier to generate. The scarce resource is no longer the update. It is the trust required to act on it.

Reinvest the time AI gives back

The practical opportunity is not simply to use AI tools. It is to redesign where a program manager's time goes.

I find it useful to divide the work into two categories.

1. Information work includes summarizing meetings, drafting routine communications, compiling updates, maintaining logs, and producing a first pass at analysis. AI can handle or accelerate much of this work.
2. Judgment work includes diagnosing why alignment is drifting, structuring a decision, interpreting weak signals, coaching a contributor, resolving a tradeoff, and building trust with a resistant stakeholder. This is the work that deserves the time AI gives back.

The trap is easy to fall into. A program manager saves three hours on status reporting and uses those hours to produce more dashboards, longer documents, or additional meetings. The person becomes more efficient, but the program does not become better.

Instead, AI should be used deliberately:

- Automate predictable information work where the inputs and output are easy to verify.
- Use AI as a thinking partner that challenges assumptions before stakeholders do.
- Improve the quality of first drafts, risk scans, and decision preparation.
- Keep accountability with the human who understands the context and owns the consequences.

AI can sit at the table. It does not get the vote.

Conclusion: The role is not disappearing. Its value is moving.

AI will not remove ambiguity from complex programs. It will not make competing priorities disappear, give every team the same incentives, or eliminate the need for difficult decisions. It will simply make the production and movement of information much faster.

That changes the standard for program management. When the status report can draft itself, the value is in knowing what the report means. When the risk list is generated automatically, the value is in recognizing which risk deserves attention. When the options can be summarized in seconds, the value is in creating the trust and clarity required to choose.

The most valuable program manager in the AI era will be the person the organization trusts to turn incomplete information into a responsible decision, not the person who produces the most information.

Author's note

This article develops ideas first presented by the author in "The AI-Era PM: Becoming a Strategic Technology Leader," delivered to the PMI San Francisco Bay Area Chapter in July 2026. The views expressed are the author's own and do not represent any organization.

AI use disclosure

Generative AI was used to help with grammar, sentence structure, organization, and clarity. The ideas, arguments, examples, data points, and conclusions in this article are my own and are drawn from my firsthand professional experience. All workplace

examples are based on real situations, with certain identifying details generalized where appropriate.

Reference

Gupta, A. (2026). The AI-Era PM: Becoming a Strategic Technology Leader [Presentation]. PMI San Francisco Bay Area Chapter, California, July.

About the Author



Ankur Gupta

California, USA



Ankur Gupta is a Senior Staff Technical Program Manager working in AI infrastructure at LinkedIn. He has more than 15 years of experience leading complex, cross-functional technology programs across machine learning infrastructure, distributed systems, reliability engineering, and large-scale platform transformation. He is an IEEE Senior Member, published author, and speaker on AI infrastructure, engineering execution, and the evolving role of technical program management.

Ankur is based in San Jose, California, USA. He can be contacted via LinkedIn at [linkedin.com/in/agupta03](https://www.linkedin.com/in/agupta03) and found online at ankurgupta.me.