



ARTIFICIAL INTELLIGENCE | Partner Insight

Four Ways to Redesign Entry-Level Jobs in the Age of AI

by
Jacob Clemente



6 Min Read

July 24, 2025

Over the past few months, a growing number of voices have raised concerns about generative AI's impact on entry-level jobs. And yet, there's been less attention paid to what companies should be doing about this, such as how to redesign those roles.

Deciding not to hire junior workers anymore isn't a winning strategy. "Firms that do that as a knee jerk and default reaction are just going to lose eventually," says Matt Beane, associate professor of technology management at the University of California, Santa Barbara and author of *The Skill Code*. "You have to find a way to grow the next wave of senior people, and also there are cases in which junior people are damn better at all this stuff than senior people are."

Junior workers can act as a "cultural refresher" for organizations, which can get stuck in the old ways of doing things, explains Kian Katanforoosh, founder and CEO of AI startup Workera. He advises looking for junior "AI-savvy, AI-first risk takers" with a mindset geared toward breaking things.

There's also an economic case for continuing to hire entry-level workers. Cormac Whelan, CEO of PDF and e-sign company Nitro Software, explains that his company can hire people out of college for significantly less than what it pays for people with three or five years of experience. But those recent college grads can quickly learn a lot when you sit them alongside the company's more experienced engineers, he says, and "I now have a much more economically scalable model."

Finally, AI tools can amplify the value you get out of entry-level workers. At PwC, for example, associates are using AI to speed up a range of different tasks, such as year-end reviews of 10-Q and 10-K filings, says Margaret Burke, the firm's talent acquisition and development leader. One associate working on a technology-modernization project for a client used AI to help produce a customized 80-page transformation guide, helping the team meet their tight deadline.

AT A GLANCE:

- A framework of four types of entry-level jobs in the age of AI emerged, including jobs where you have a competitive advantage, more difficult work, and creating entirely new roles.
- There's an economic case for continuing to hire entry-level workers, but they can quickly learn a lot when you sit them alongside the company's more experienced employees.
- Research suggests AI tools can uplevel less experienced workers in many areas, enabling them to perform more complex work faster.

So what can companies that care about developing the next generation of talent do, when AI seems poised to disrupt their current tasks? Shift the focus from whether entry-level jobs will survive to how you can redesign them such that they're adding more value. We spoke to a range of researchers and practitioners about how to do that. Here are four frameworks and tactics that emerged from those conversations:

Create entry-level jobs where you have a competitive advantage.

Guild CEO Bijal Shah's advice is for companies to home in on their competitive advantage, figure out how workers can unlock it, and create entry-level jobs there. The advice is less geared toward AI specifically and more focused on creating junior roles in areas where they can learn important skills and where you know you'll need talent for the foreseeable future.

Shah gives the example of a company that prides itself on being customer-centric and hires junior workers into roles where they can directly interact with customers. "That builds their ability to better understand the customer, to understand what the customer's problems are." Similarly, if supply chains are a competitive advantage for a firm, it might create entry-level jobs that allow junior employees to gain hands-on experience with sourcing, logistics, and operations.

This approach ensures you're developing a bench of talent who really understand the business and how to set it up for success. It also creates jobs that are relatively more future-proof at your company, as you'll continue to need people who know how to make your company stand out from competitors who are using the same tools as you.

Create entirely new roles.

In many cases, AI adoption may create the need for new types of work that you can slot entry-level workers into. IBM, for example, has an AI tool for HR that answers employee questions. Employees can rate the quality of the response, and when someone gives it a thumbs down, one of the company's entry-level HR employees contacts them to ask them how their experience could have been improved.

"That's now an entry-level job to seek out feedback from employees real time," Nickle LaMoreaux, the company's CHRO, said recently. That role also monitors broader trends, noticing, for example, if the system has recently gotten a particularly large number of 401K-related questions. In that case, the person would contact the benefits team to figure out the underlying cause and resolve the issue.

Create ways for junior workers to perform the higher-value tasks that their more experienced peers do.

New research by Arvind Karunakaran, assistant professor of management science and engineering at Stanford, and his co-authors shows what this looks like in practice for roles that are lower on the org chart.

When studying AI implementation at a law firm, the researchers found that paralegals in one of the divisions (let's call it Division 1) used and experimented with the company's AI tool significantly more than their counterparts in another, very

similar division (Division 2). One key difference was that the manager of Division 2 framed the benefits of genAI in terms of productivity improvements. The manager of Division 1 framed the benefits of the AI tool in terms of how it could make the paralegals' jobs better, by automating tasks they hate doing and giving them opportunities to try tasks they've always wanted to do with the freed-up time.

As a result, the paralegals in Division 1 over time experimented with the tool more, even trying it out for new tasks they had less experience with, like legal research. Importantly, Karunakaran tells us, the manager in Division 1 followed through on their commitment to improve the paralegals' jobs. In addition to providing them more training on how to use the AI tool and slack time to experiment, they gave them training and mentorship opportunities for tasks they wanted more exposure to, like legal research. Karunakaran and his co-authors found that over time, the paralegals started to take on more complex tasks.

Karunakaran tells us that this shift initially created some tension between paralegals and the firm's junior attorneys, who felt that the former were encroaching on their work. The manager's solution was to then shift the tasks the junior attorneys spend their time on further up the stack to more impactful work, like getting them involved in client conversations earlier, he explains.

Ramp up the difficulty of the work you assign entry-level workers and keep a senior person in the loop.

What if companies leaned into that and increased their expectations for junior workers? This approach differs from the previous one mainly in that it's more top-down and puts workers on a more accelerated learning path.

UCSB's Beane is currently conducting a study on the use of generative AI on software development teams at a range of companies. The study is ongoing, but some insights are already starting to emerge.

One "not-too-common, but interesting" practice he's hearing about involves managers or senior developers assigning the junior developers in their group problems that are wildly more complex than what they were previously assigned in a shorter amount of time. Once the junior developers are done, their code gets tested and the junior developer gets feedback from a senior colleague. "We're going to focus on the areas that you are unlikely to have noticed. You'll have produced working code that is terrible in some way. I'm going to show you where those three terrible parts are and then tell you to go fix it."

This approach involves unique risks, "but it does seem to be yielding fruit in the study," Beane tells us, adding that it can significantly boost the productivity of the group as a whole, even if it means the more productive senior person is now spending less of their time writing code. It's also a great opportunity for junior developers to learn at a higher level, and they seem to be motivated by this approach.

“There are cases in which junior people are damn better at

all this stuff than senior people
are.

- Matt Beane, associate professor of technology
management at the University of California, Santa
Barbara

It does have its downsides, though. Beane says that he's heard from some senior developers that it can be very annoying to review code from junior employees that's full of errors. Others see it as an opportunity to coach junior colleagues. "I think that'll be a crowd splitter," he says. "It's not obvious from a distance who's going to enjoy and want to move into that role of supervising people who are aggressively supervising clouds of agents trying to get stuff done."

Although Beane's research focuses on software development, the approach of giving junior workers significantly more challenging problems and having a senior colleague coach them on what they got wrong and why is a practice that can be applied to many other professions.

This research is ongoing, and the definitive best practices will become clearer over time. But this is an approach worth experimenting with. "If the world was listening live to this conversation and got no further information from anyone and they had to go start to get to work, the answer is just go hard and way more than you think you reasonably could achieve," says Beane. "Be much more aggressive than you're comfortable with, as long as there's a senior person backstopping."