

6

BEST PRACTICES FOR ADOPTING A LOW-CODE STRATEGY

Low-code and no-code strategies have the potential to accelerate government software development projects, but they need to be handled with care.

How should agencies make the best use of these tools? That was the focus of a recent [GovLoop virtual event](#) featuring Ravyn Manuel, senior application developer/architect and a DevOps engineer at the Smithsonian Institute's National Museum of African American History and Culture. Here are some highlights of that discussion.





1

KNOW THE DIFFERENCE BETWEEN LOW CODE, NO CODE

Low code and no code are both development platforms that allow users to build applications with limited or no coding knowledge, but they are not the same thing.

With low code, “there is an expectation that you will have a little bit of coding experience — maybe not a lot, but you are familiar with coding platforms, or just coding in general,” Manuel said. “No code is exactly as it says: You don’t have to have any kind of coding experience.”

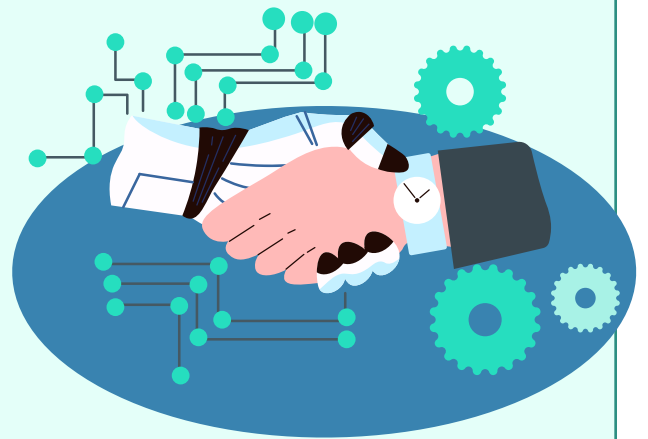
Different users will utilize these tools, depending on their level of experience. An IT professional might lean toward low-code, while no-code “is for people who just want to get it done fast,” she said.

2

ALIGN THE TECHNOLOGY TO A BUSINESS PROBLEM

A good starting question for low- and no-code approaches: “What is the actual business problem? What is it that you’re trying to solve?” Manuel said. “Get that business problem from all perspectives.”

Once you’ve defined the scope of the problem, align to the approach that fits best. Don’t just follow the buzzwords around low- and no-code alternatives. “Make sure you understand the technology that you are even thinking about,” she said. “Even if you’re not a technologist or an IT person, there’s always documentation to tell you [at a] high level what that technology is going to do for you.”



3

MANAGE THE RISKS

A number of caveats apply when going the low- or no-code route. Here are some common pitfalls:

- Users can introduce security risk by skipping needed testing steps in the push for rapid development.
- Low-code solutions might not integrate into existing large systems, or bringing those systems together could inadvertently expose personally identifiable information.
- Low- and no-code solutions may deliver limited functionality. For example, a low-code solution might not meet the needs of a neurodiverse user.

All these are manageable risks, Manuel said, but they require awareness and a thoughtful approach.

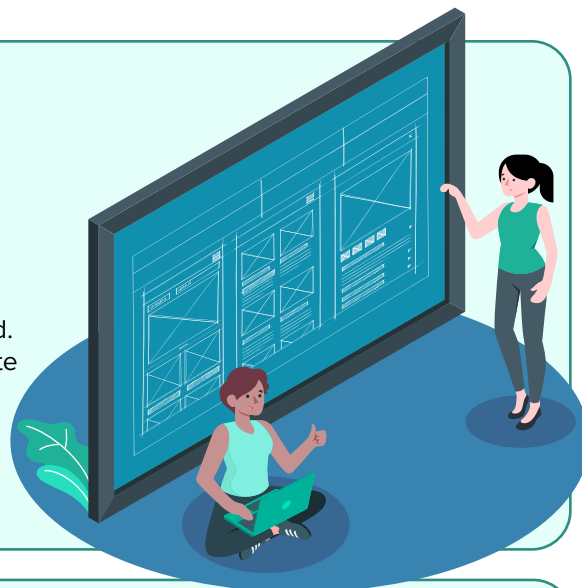


4 START WITH A PROTOTYPE

As with any new IT strategy, it makes sense to start small and practical with low- and no-code, in order to demonstrate the possibilities.

“You have to show them, and you don’t have to be extravagant about it. You have to give a use case, but a use case that’s relevant,” she said. Then you can leverage that use case to focus an initial effort and create a working prototype to bring low-code and no-code methods to life.

“Don’t try to overhaul the whole system,” she said.
“You have to start small.”



5 LOOK SHORT- AND LONG-TERM

Developers typically will turn to low- or no-code approaches to meet an immediate need. That makes sense, but at the same time it’s important to be developing a longer-range strategy.

“I’m an uber-proactive person,” Manuel said. “I never focus on tools: I focus on the underside. Can I take my data from here very easily? Because the data isn’t going to change. Can I migrate that over to whatever the next platform is, or the next solution?”

When bringing in a new capability, “I am always looking five or six years out,” she said. By keeping in mind those long-range questions, even while tapping the tools to meet short-term goals, IT teams can maximize the value of their investment in low-code and no-code capabilities.

6 FOCUS ON THE EMPLOYEE WORKLOAD

When considering low- and no-code approaches, it’s important to keep in mind the main goal, Manuel said. The point here is to relieve stress on an overtasked government workforce — to make their lives easier.

“The government workforce is not getting a whole lot bigger right now. It’s not like you’re doubling the size of your teams. But the mission of your organization isn’t changing,” she said. “These solutions can offload some of those tasks and make the jobs a little easier for the employees that are there.”

The success of a low- or no-code effort can be measured not just by how quickly the development cycle moves along, but also by the impact on those involved in the effort. Process improvement in development ought to lead to quality-of-life enhancements for those who make government run.

To learn more about the ins and outs of low code, watch the full session [on demand](#).

