accenture

# LOW CODE PODCAST

## AUDIO TRANSCRIPT

**Adam Burden** [00:00:00] But the real magic here is what I'll call the democratization of technology.

**Penelope Prett** [00:00:09] Hi, everyone, it's Penelope Prett, Accenture's Chief Information Officer here. I'm happy you can all join us today as we speak with Adam Burden. Adam is the center's technology lead from North America. And today with Adam, we're going to talk about low code capabilities and what they mean for organizations. Adam is going to share some of his expertise about what advantages low code can bring to companies and what they need to be thinking about as low code begins to enter more and more into our everyday vocabulary in information technology departments. Welcome, Adam.

**Adam Burden** [00:00:40] Thanks, Penelope. You know, this topic is one that I've been following for a number of years, and it's really exciting to see it on the forefront of many of our technology leaders minds, because we've  had this conviction for such a long time that it's going to be an important field. I couldn't be more excited to have an opportunity to chat with you about it and share my enthusiasm for this space.

**Penelope Prett** [00:01:05] Super.

**Penelope Prett** [00:01:06] So let's jump right into it then. We're hearing a lot of people throw around the term low code today. Right? Can you start by just explaining to the audience a little bit about what exactly low code means to you and how it works?

**Adam Burden** [00:01:19] This is a really good way to start this discussion, because I do think that there are terminology problems out there. First of all, we have started breaking this down into what we'll call three categories of code. We have what we call pro code, which is exactly what you think it is. It's code only, right? It's people, software programmers that are writing applications, but in code, and that is professional developer territory. Then you have this middle ground, which is what we call low code. Low code often has a little bit of code in it, it doesn't necessarily have it (but I'm going to talk about what I mean by that in a second). It usually has models, metadata and use processes of forward engineering to generate applications. Sometimes it's called scaffolding as well, like when you're assembling a building or something like that, you use the scaffolding. It's a way for you to say very quickly this is what I want an application to do, it will assemble all the parts around it that you really need. Then you go in and fill in the middle with a little bit of code and some of the tools don't require it at all, which brings me to the third category, which is no code.

There is also what we'll call total codeless development with no code, where people are only using models and metadata to build things. I'd say, when you think about people who are in this category of citizen developer, they're the business users, that don't necessarily have a technical background, that can pick up these tools and be able to use them to create productive applications for their department.

So, we have really three categories out there, and that's how we really define this low code area –pro code, low code, which is a little bit of code and then no code at all.
It is just a generation of applications that the citizen developer can take on.

In that middle category, low code, it's kind of like a hybrid use model. We have folks from all walks of life that work in low code because they can choose the degree to which they actually have to write software inside of the (application). Sometimes it's very little, sometimes it's a lot, but it's a way for them to accelerate development. That’s a little bit about the definition of what low code is all about.

**Penelope Prett** [00:03:34] Thanks for clarifying, and while you're on it, can you talk a little bit about why low code's popularity is exploding right now? What are the competitive advantages they can offer to a company?

**Adam Burden** [00:03:45] This is something, it's funny, we've been predicting this for years. Right? And I don't know, I have a bad habit of predicting things way too way too early that are going to happen. We have definitely reached that now with this area. And the question is why? What's occurring now that has really made this so popular?

I think there is a couple of factors. One is you've seen this explosion of agile development and the pace of technology moving faster than it ever has in history. You also see this adoption by the business side of organizations who are saying, “why can't I get application changes or do things faster? You know, in a way that's going to allow me to enjoy the benefits of these systems much sooner than what we’d see from traditional projects.

It's in that perspective, that belief in those organizations, that has really spawned this desire to be able to do citizen development or sometimes called DIY development, where they can build things themselves. The real magic here is what I'll call the democratization of technology.

I can give you a great analogy for this, if you look back at the history of artificial intelligence, it was the domain of extremely deep pocketed enterprises for a long time. It required incredible amounts of investment, curation of very expensive data sets to be able to even teach the smallest of models on recognition of images, for example.

What we’ve seen, let's say, the last couple of decades, maybe the last decade or so, has been a democratization of AI. The cloud providers began to make these AI tools and maybe more so, the compute resources available at the fingertips of people who were working in the artificial intelligence field. No longer did they necessarily have to have a deep pocketed investor or a university apparatus for them to be able to do AI work.
It spawned this this incredible era of innovation that we've seen in the A.I. space. I mean, entirely new thinking about models and things like adversarial networks. That’s what we're seeing happen right now with cloud and the development of cloud applications. Low code is something, as I said earlier, that’s been around for a while. We've been watching it.

Now what we're really observing is that this democratization of cloud and the application development space is creating this explosion of innovation. It’s innovation that you're see in terms of the number of tools that are available and the domain specific languages. We have the hyperscale like Amazon, and their honey code solution, we’ve got independent companies like Mendix, Outsystems and others,

They’ve all started to make their tool sets available in the cloud and it's at the fingertips of people to be able to use. Every enterprise knows and has really embraced this journey for them to be more digital. They want to be able to take advantage of these systems because what they want to do is take their employees and enable them to automate the ordinary and unleash the extraordinary. By putting that innovation at their fingertips, they're enabling them to do more than they had ever in the past.

I really think that the democratization of these technologies which have really made that difference. We're going to continue to see this, Penelope. I think that we're really just on the edge of something here. With the tools that are available right now, they're quite varied, they're all over the map. It's like what we saw with platform as a service several years ago, where there were hundreds of different tool sets and it's now a consolidated down to maybe a dozen or so of really popular ones. I think that this space is going to be like that for a while.

It's up to companies, though, to enable their employees, to be able to use these tools in the right way. I think this popularity is just going to continue to expand and grow because many of them feel like that they need this innovation to be competitive in this in this world of digital enterprises. And they're right. Low code and low code, no code type solutions are absolutely a path for them to get there. That's been driving the popularity.

**Penelope Prett** [00:08:19] Well, Adam, what you're saying is really ringing my bell, because when I think about what's happening inside our own Accenture house, we're using the power of Microsoft platform tools like Power Platform to help extend people's workspaces, to come up with new ways to connect, to fill niches of functionality that can be difficult to reach from other big packages. But it leads you to kind of an interesting question to ponder. Right?

Which is what is actually the impact of these low code tools going to be on packaged software applications and SaaS providers? How do you think their worlds will interact?

**Adam Burden** [00:08:53] OK, well, that's for another podcast, maybe, Penelope, because I've actually done a whole another series of things around something I call the buy versus build equation that is changing, but let me comment on that a little bit later.

First of all, I think most of the major software organizations, pick on ServiceNow, SAP, etc. they've started to build more low code type tools inside of their solutions. SAP's Cloud platform, Salesforce, has done this with Lightning and others. Then we also have the MaGg providers starting to do things. You just mentioned Microsoft and what they've done with Power platform. We've got Amazon, AWS, that has been doing some things with Honeycode and Google's with App sheets.

These are going to continue to expand in popularity and further availability to lots and lots of different entities that are out there, and they'll even do it at educational levels as well.

The thing that I think is really interesting inside here, is that companies are now looking at this, and why I am saying that the buy versus build equation is changing, they're evaluating for themselves, what are things that really make us competitively different in the market? What are the things that enable us to move more rapidly and capture more business, delight our customers, our business partners, our employees? They're trying to figure out what are the things that are more customer facing of those applications, more employee direct facing, sometimes referred to as the systems of engagement that are out there. These systems of engagement, they're not the human capital management systems, not the systems which are the financial and back office systems. There's not a lot of differentiation necessarily that you can get out of those.

But for the order to cash application? Absolutely, for the warehousing systems, for sure. Right?
I think the buy versus build equation is changing, which I've also written a bit of a blog about as well. I hypothesize that over time you're going to see that major software vendors are really going to open up and recognize that there's elements of their applications, that customers are just not willing to wait for them to decide that a feature is important enough for them to build and maybe release it in six months, for a year.They need to build that stuff now. They're going to open up their products to be able to expand applications.

Some of the more progressive ones have already done this. For the others that don't, I think they're going to be threatened by some of these startup type tools, who will build applications around them that allow those customers to move faster and more agile in order to be competitive.

We're going to continue to see this phenomenon come out more and more in the future. The popularity of low code, I think, is going to change the way that many enterprises look at software packages, because they may conclude in some cases that the buy versus build equation, which for so long has been tilted in the favor of commercial and off the shelf packages or cot systems, that it may make sense for them to actually build some of these things themselves.

Now, a word of caution here for really big enterprise systems. You know, you talk about things with thousands of applications. Don't underestimate the complexity in building something like that. While many of these tools, Power among them, are quite capable of handling large number of users, the sophistication of what it takes to build an ERP application or a financial system, is quite complex.

Tackle the things that are important inside. This is the message I leave you, the things that are going to give you new systems of engagement and focus on potential use cases for low code and no code. Use it in places where the software vendor themselves has started to embed this capability inside of their product. I really think that it's going to open a new era of creativity in enterprises and they're really going to feel like it's made them more competitive.

It's an interesting time to be a big software platform vendor in the space, because they need to move rapidly where they don't already have a capability like this, to enable it. Just look around. We've just seen ServiceNow make a similar announcement and I think you're going to see it from many of the other (vendors) in the near future as well.

**Penelope Prett** [00:13:35] It's an interesting vision for the future, Adam, and one which is rapidly coming on us. Right. Still pulling it back to the present.

One of the things we always like to do on this podcast series is just leave our listeners with some practical tips, everyday tips about things they need to be thinking about right now on some of these technology waves that we're getting ready to ride.

Can you share any of your practical tips on what to do today with our listener group?

**Adam Burden** [00:13:57] Yeah, for sure, I have a couple of them.

One is low code and no code are a great way enable the non-technical part of our communities that we work with to become more technical. When you think about the messages around inclusiveness, recruitment, and the scarcity of talent that's available, the ability to rapidly teach someone how to use these system and application building tools is here. It has reached a point where you don't necessarily need a computer science degree to build very functional applications.

I think that's an important lesson learned out of my experience h with this. I feel that we've really turned a corner here. One of my favorite movies, Penelope, is Ratatouille. Have you ever seen this movie Ratatouille? You know which one I'm talking about?

**Penelope Prett [**00:14:52] I have indeed.

**Adam Burden** [00:14:54] OK, so it's a really cute story, is a Pixar movie There's a chef in it, and I think his name was Chef Gustav and he used to do this television show and is "Anyone can Cook" Right? Well, I really think anyone can code.

These tools are given us the ability to do that because it's allowing people that don't necessarily have the technical backgrounds to go do that. This is an important way for you to expand your horizons in terms of recruiting from nontraditional sources.

Another angle to look at this, too, is you have many enterprises doing massive shifts, massive shifts into cloud right now. They’ve got an awful lot of legacy developers that that might have more familiarity with mainframes or AS 400s or other things. Well, I'm telling you, they're archival knowledge that they have the expertise of business processes and systems inside of that enterprise, their functional knowledge is so important.

Teach them how to use these low code solutions to continue to build and define new applications for tomorrow. It's a great way to transition existing talent that you have in a way that doesn't require them to become full-stack developers or pro coders. It is a very different world that's available now for you to do that as an enterprise, transitioning some of your workforce into cloud as an enterprise. I really think those are important lessons that are out there.

I got one more for you that I think is quite significant. Some of the early adopters in the low code no code space, Penelope, they got ahead of their skis a little bit.

The reason that that happened was their appetite for this really was quite strong, and they really released low code abilities quite broadly. Now they have an awful lot of tools that do very similar things because the standards weren't there and there wasn't a lot of controls around which ones were going to be used.

More difficult has been the security that you want to have in place around that, the guardrails around things, the reusability. If you want everybody to use the same authentication service for any applications that they're building, You need to build that into the governance for this and need to customize a lot of these tools so that there's standard frameworks that people will use and follow going forward.

Some organizations that we've been working with have taken this to a very sophisticated level. I'll credit Accenture as having done this as well, because, some of my teams have built low code applications inside of Accenture for our own purposes and we tier that and have different ranges that allows the back office person who is doing the processing. They can build their own application, but it's got controls around it. it doesn't create a security risk for us but allows them to automate many of the tasks that they otherwise would have performed manually in the in the past.

Think about that as you're rolling this out in your enterprise. Think about how putting the right governance and guardrails around this. One of the best thinkers in this space, one of I really admire him, he always says it's best to go slow, to go fast when it comes to low code. And that's very true.

Roll this out a little bit, get your governance and processes right, and then you can go super-fast. But if you try to go too fast, it's just going to make you come to a grinding halt.

These are a couple of great lessons for you. Go slow to go fast and everyone can code courtesy of Ratatouille.

**Penelope Prett** [00:18:33] Well, Adam, I thank you so much for coming today and sharing your thoughts around this whole low code environment. It's been fascinating and enlightening for me, and I've learned a few new terms like scaffolding that I'll be using going forward. I'm sure our listeners got a kick out of it as well. So, thank you for taking the time and it's always a pleasure to speak with you and hear what's the latest thinking.

**Adam Burden** [00:18:53] I really I hope that people really, when they're listening to this podcast, that they really take to heart the enthusiasm that we Accenture, me, myself personally have for this space and the innovation that's going to be possible as enterprises take on low code and no code.

it's really unleashing an entirely new world of possibility. It is exciting times ahead. Thank you so much for giving me a chance to talk about it. I look forward to having a chat with you again soon. Take care.

**Podcast Close** [00:19:25] Thank you for joining today's podcast. Be sure to subscribe to the Accenture CIO podcast series on Apple podcasts or Spotify. Find the full CIO Podcast audio 24/7 podcast series and additional ways to subscribe at Accenture.com/CIOPodcast.