Neil Ward-Dutton

Low-code platforms – ideal to drive rapid digital transformation

June 2017

Low-code app development platforms have seen a huge spike in popularity over the past couple of years. Their immediacy means they're a great fit for situations where you need to start small and make a difference quickly. However low-code platforms aren't only for small-scale, departmental apps that solve team co-ordination problems. Low-code platforms can also be very effective in turbocharging the efforts of enterprise technology development teams in addressing mission-critical business issues – making them more productive and more able to make changes fast, with confidence. Not all low-code platforms are created equal. The trick is in knowing how to spot an enterprise-ready low–code platform.

MWD Advisors is a specialist advisory firm which provides practical, independent industry insights to business leaders and technology professionals working to drive change with the help of digital technology. Our approach combines flexible, pragmatic mentoring and advisory services, built on a deep industry best practice and technology research foundation.

This paper has been sponsored by Appian

Digital transformation demands a new business-technology contract

Every organisation with a digital agenda has its own take on the particular priorities that make most sense. Your organisation might be primarily focused on building new online channels for products and services, digitising business operations, giving field-based employees better information and tools, or creating new digital products or services – or perhaps a combination of these (or other things).

Whatever the particular priorities driving your agenda, one crucial piece will be part of the puzzle: how to use digital platforms and tools to 'level up' in driving faster business and technology change – and delivering real business results more quickly.

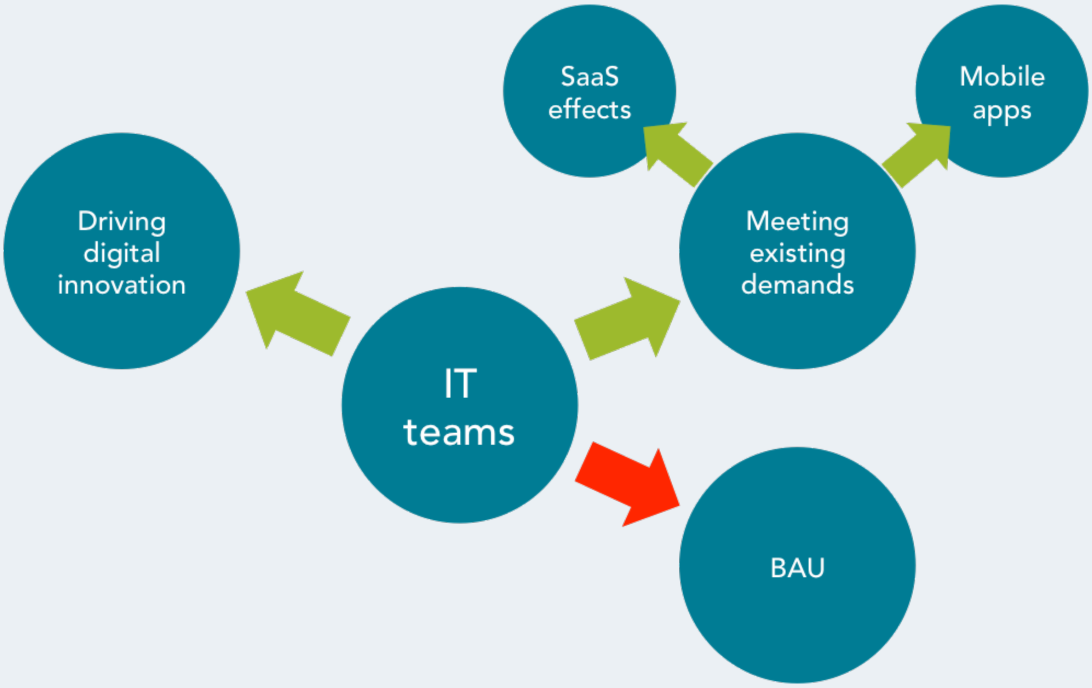
How we get from here to there, though, is something that a great many organisations struggle with. The answer is partly about the organisation itself, and partly about the tools it has in its toolkit.

Relentless pressure on IT teams

It's all very well to talk about how digital technologies enable organisations to drive business-technology change in new ways, but we know that a great number of CIOs, and the IT organisations they manage, are under relentless pressure to meet their existing commitments and expectations. There's little 'headroom' in the technology resources available to most organisations, so that makes any thought of real transformation very difficult indeed to prioritise.

In MWD Advisors' Spring 2017 CIO survey[1], one of the most striking findings was that although 81% of CIOs expect to be taking more active roles in digital transformation and innovation initiatives over the coming 24 months, 69% also say that their own teams struggle to keep up with demand for new technology capabilities (and changes to those capabilities).

Figure 1 Relentless pressure on IT teams



Source: MWD Advisors

In our experience this demand partly springs from a rise in net new business requirements (in large part for mobile business apps); but it's also being driven by the implications of fast-evolving technology landscapes (see figure 1).

In particular, organisations commonly struggle with the implications of the proliferation of easy-to-acquire, easy-to-use SaaS applications (primarily to support sales and marketing activities, but not exclusively so). New centres of gravity for technology spending are appearing, and in these centres 'standard' IT concerns (relating to security, integration, data quality, compliance and so on) are easily, even if not deliberately, overlooked.

The answer: collaborate, don't just serve

We see two common types of response to the challenge of managing demand in this new, uncertain, fast-moving world where SaaS-centred investments are so easy to make (and where, by the way, overall IT budgets don't tend to be growing by much):

- **Resistance and isolation**. Some CIOs and their teams attempt to use a mixture of persuasion, influence and force to dismantle what they see as unnatural centres of (SaaS-centred) technology procurement and use that are outside their control. Some, with board-level support, succeed: but the challenge is that 'shadow IT' is a symptom of the underlying 'new normal' in business technology – it's not an isolated phenomenon. Because the underlying 'new normal' environment can't be dismantled in any rational way (at least, not without sentencing the wider organisation to a severely handicapped future), any victory over it turns out to be temporary. And once the 'new normal' reasserts itself, the IT organisation's reputation is even more tarnished.
- **Acceptance and collaboration**. This is a far more sustainable route to take, though it requires maturity (for IT leaders to truly accept that they cannot control spending on, or use of, IT). Instead of fighting against 'shadow IT' centres, this route involves embracing all groups that need IT resources and capabilities, and working collaboratively with them to support them in their efforts, highlight potential risks and controls, and thereby demonstrate the value that a joined-up approach can bring.

An 'acceptance and collaboration' response is fundamentally about creating a new kind of contract between business and IT: a contract that revolves around joint working to serve the needs of customers and other stakeholders, rather than revolving around IT acting as a straightforward service provider to business teams.

However by itself, even the most open and positive approach to managing relationships with all sources of technology demand will not make the problem of managing demand go away, though it will make things easier. New SaaS-centred investments still need to be integrated and governed effectively, and demand for new app extensions and service features (for customers, partners, suppliers and employees) needs to be met.

What's also needed is a technology platform that truly supports collaboration between technologists and other businesspeople in meeting that demand, and that enables those groups to work together quickly to get things done.

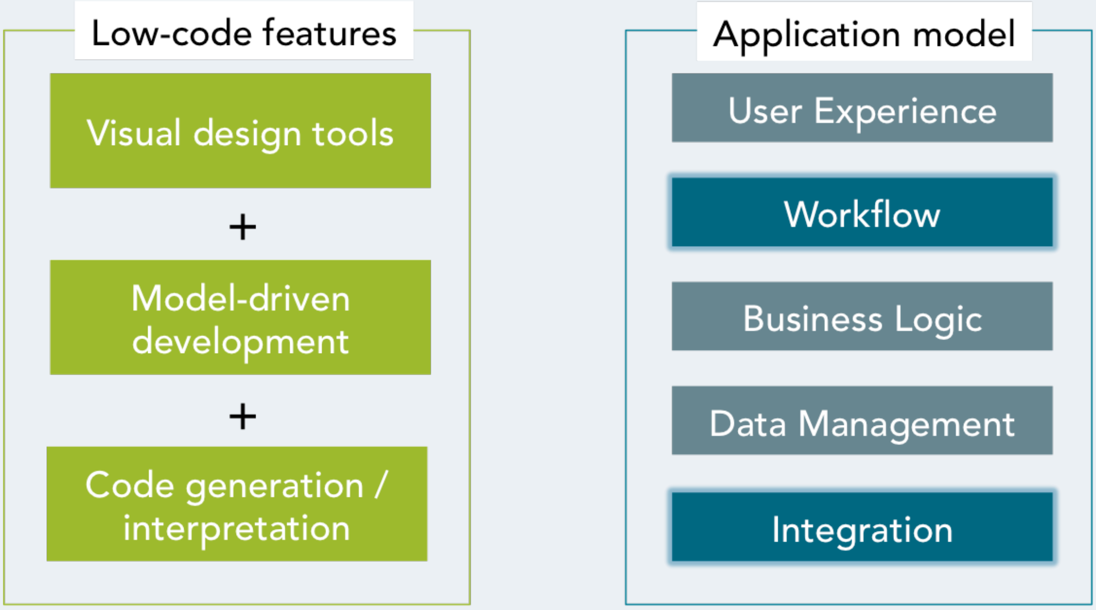
This is where 'low-code' application platforms fit in.

The low-code (r)evolution

What is a low-code application platform?

Low-code application platforms allow teams to build business applications more quickly, and with less effort, than using 'traditional' application development tools and languages. They typically do this by employing a combination of visual design tools, 'model-driven' development (which enables people to specify application behaviour through high-level configuration rather than detailed coding) and code generation or interpretation (to create the application code that runs behind the scenes). See figure 2.

Figure 2 Low-code platforms: speed, collaboration, agility



Source: MWD Advisors

Most low-code application platforms are built around application models with three distinct layers:

- A layer focused on data management (where you specify the data your application will manage).
- A layer focused on business logic (where you specify the core behaviour of your application).
- A layer focused on user experience (where you specify how people will interact with your application through screens, forms, menus and so on).

Some low-code platforms are more sophisticated, and also provide application model layers for specifying automated workflows, data and service integration components, and more.

Low-code application platforms are crucial enablers for digital transformation and innovation because their use of model-driven development techniques and visual design tools make them a great fit for blended teams of business and technology specialists to collaborate and create agile applications quickly.

Back to the future

This is not the first time 'low-code' approaches to application development have been part of the business technology landscape. In the 1990s a clutch of 'rapid application development' (RAD) tools gained huge popularity: if you're old enough to have used Microsoft Access, Visual Basic, Delphi or PowerBuilder you'll already know the basic principles of low-code platforms.

The 1990s wave of low-code platforms became popular in response to a new kind of business computing platform becoming popular – networked personal computers with GUIs, connected to relatively small, cheap, networked servers running relational databases. This drove demand that most IT departments, still staffed and resourced to manage mainframes and proprietary midrange systems, couldn't easily fulfil. In addition, a rapid expansion in the variety of new wave of client and server operating systems and DBMSs available meant that for many large organisations, the challenge of developing applications with traditional tools that would work well everywhere was just too great.

This story is being retold today, although some of the details have changed. Today, the new business computing platform revolves around cloud-based applications and platforms (servers) and mobile apps (clients) rather than on-premises workgroup servers and PCs. Today, organisations' need for low-code tools again springs from the desire to deliver applications that work across heterogeneous, distributed environments; but we have the added complication that the speed of change – from a business competition perspective as well as a technology perspective – is markedly faster than it was in the 1990s.

The bigger picture, and why a strategic choice is important

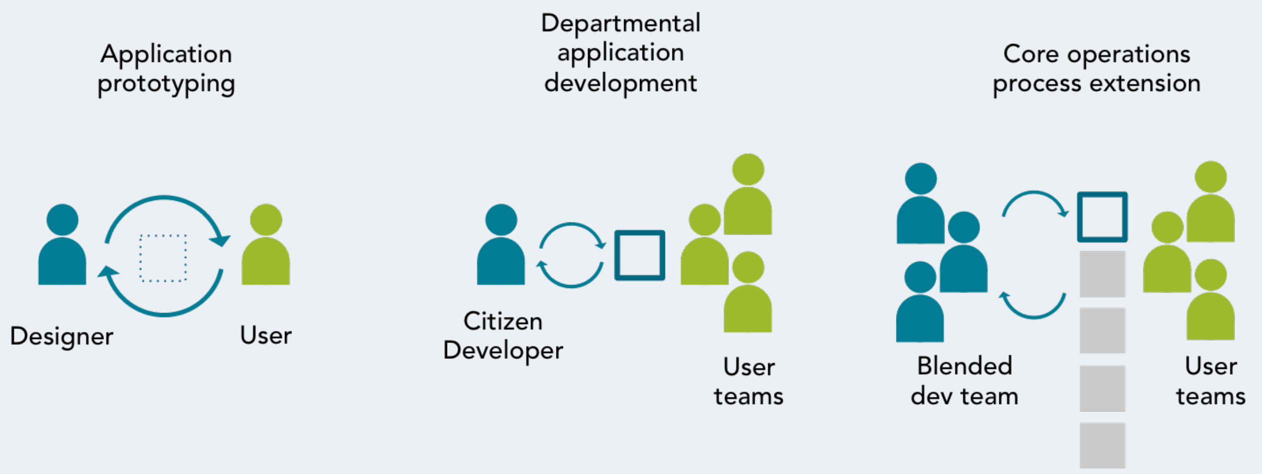
So if low-code application platforms are crucial enablers for digital transformation and innovation, how do you choose the right one for your organisation, when there are dozens of platforms to choose from (with more choices arriving every quarter)?

Understand the three modern low-code platform use cases

It's not obvious, but there are three distinct (but related) types of use case and three value propositions for modern low-code application platforms in relation to digital transformation initiatives. If you think you're in the market for a low-code application platform, it's crucial that you understand this – and use this knowledge to guide your choice. The three use cases (shown in figure 3) are:

- **Application prototyping (P)**. Digital innovation teams can use low-code application platforms to iteratively create and enhance 'minimum viable products', working in collaborative environments face-to-face with customers and other stakeholders. Here, the principal value of low-code application platforms comes from their ability to enable developers to 'work out loud' in front of stakeholders, building and demonstrating features visually. Platforms don't have to be super-sophisticated to support this use case, but they do have to be quick.
- **Creation of standalone departmental applications (D)**. Teams of 'citizen developers' (businesspeople highly familiar with their own area of work and with analysis skills, but likely not trained in software development tools or techniques) can use low-code application platforms to create applications that solve departmental pain points – often, by replacing the *status quo* of having teams share spreadsheets via email. Here, the principal value of low-code platforms comes from their ability to make it easy for relatively non-technical people to build relatively straightforward business applications that can manage data and automate workflows. Platforms don't have to offer integration capabilities to support this use case; the most important characteristic here is ease-of-use.
- **Extension and augmentation of operations processes (E)**. Specialist IT practitioners can use low-code application platforms to extend the scope and capability of existing automation across operations processes, working in collaboration with subject-matter experts and citizen developers. This use case often features the need to create new mobile apps – either to enable new groups of staff with operational responsibilities, or customer groups, to participate in automated workflows in new ways. To support this use case effectively, integration capabilities are likely as important as speed of development and delivery.

Figure 3 Three use cases for low-code platforms in digital transformation



Source: MWD Advisors

As you can see from the explanation above, the value of low-code application platforms – and the capabilities that come to the fore – is subtly different depending on the use case in question.

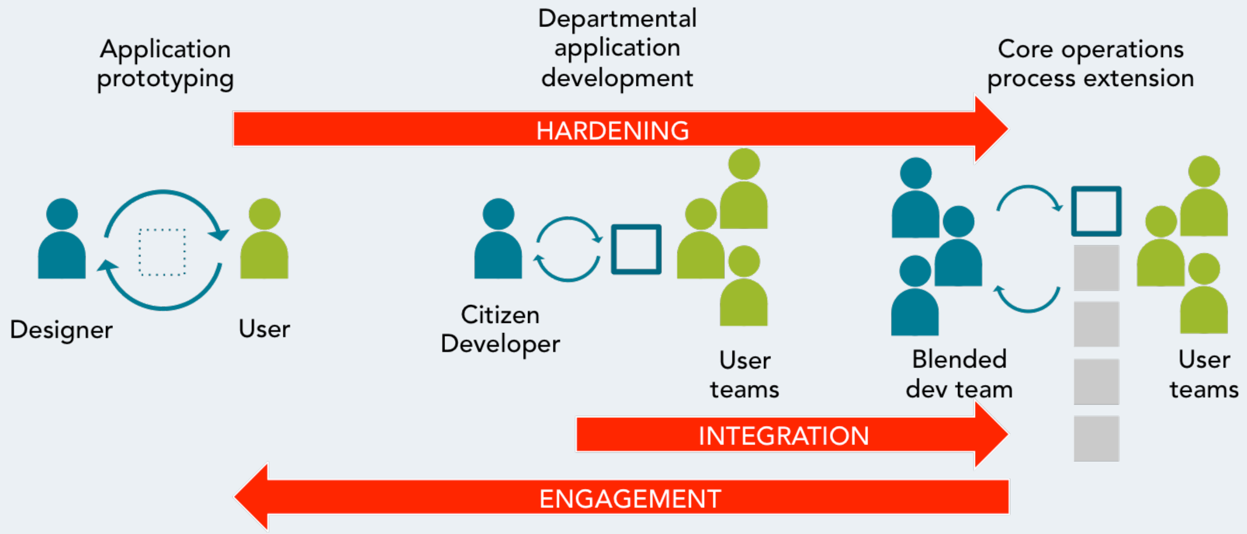
In digital transformation initiatives, you have to work across use cases

But there's more: if a low-code application platform is to truly support digital transformation, it has to support *all* these use cases – P, D and E.

Specifically, as figure 4 shows:

- Application prototyping exercises (P), when they lead to successful propositions, need to be 'hardened' and transitioned so that they become part of the business-as-usual operating environment. This requires the implementation of use case (E).
- If departmental application development exercises (D) are going to deliver truly sustainable and incremental business improvements over time, organisations will need to integrate applications that were initially built standalone with other applications – again, requiring the implementation of use case (E).
- When extending core operations to address new opportunities to engage customers or new staff groups, it will often make sense to use modern digital innovation practices that promote rapid, iterative prototyping and user feedback. That's where use case (P) plays.

Figure 4 How low-code use cases link together



Source: MWD Advisors

If you don't take the interconnectedness of these use cases seriously, you run the risk that you'll end up with a platform that might be fine for prototyping (P) and/or addressing small-scale team co-ordination challenges (D) – but no good at getting at the deeper challenges that organisations are facing. Or vice versa: a platform that's great as a 'hardening' platform (supporting use case E) but fairly useless in support of use cases P and D.

The openness imperative

We've talked already about how digital transformation is fundamentally about a new model of change: and it's important to understand that changes you'll have to deal with will not be confined to the implementation of business logic and policies. Over time it's likely you'll need to deal with change in, among other things:

- The resources your applications need to integrate with.
- The ways that people (and other systems) will need to access the logic and data your applications present.
- The places your applications (or elements of them) need to run.

Platform openness is the key here – and our recent CIO survey (referenced at the start of this report) reinforces its importance: 69% of our survey respondents either agreed or strongly agreed with the statement "Technology openness and flexibility are more important to us than feature lists".

Netting it out – requirements for low-code application platforms

To pull all these threads together, we've assembled a table of features and capabilities that you should assess when selecting a low-code application platform to use within a digital transformation initiative. This is not an exhaustive list, but it highlights some key features and capabilities that may not be immediately obvious.

Feature / capability	Description	Reason
Separation of concerns	The platform's architecture clearly separates work that non-specialists can do in building applications (for example specifying business rules and high-level business processes, and designing user interface elements) from work that technologists may need to do (integration development, technical process and UX configuration, etc).	Collaboration within mixed teams to deliver fast, at scale.
Team working	The platform provides (or integrates with) a facility for individuals to work from a shared repository of specifications, and check in / out elements to protect others from overwriting changes.	Enable multiple individuals to work on multiple projects at once.
Change management	The platform provides (or integrates with) a facility enabling teams to package up work, store multiple versions of packages, assemble tested package versions to create point-in-time applications, and reuse packages within and across projects.	Deliver accurate change, at pace and scale, sustainably.
Automated testing	The platform makes it easy for quality teams to build and run automated test suites to carry out functional, component and integration testing.	Integrate low-code work into broader IT application quality lifecycle (particularly use cases D and E).
Open APIs	The platform offers open APIs to enable developers to control the behaviour of applications from external services, embed application functionality, build new user interfaces, and so on.	Make it easy to adapt applications to environmental changes.
Mobile app framework	The platform makes it easy for application functionality to be delivered to popular smartphone and tablet devices.	Make it easy to extend applications out to mobile workers and customers.
Integration	The platform makes it easy for new application functionality to be robustly integrated with existing applications, data sources, business processes and systems.	Use case E.
Rapid prototyping	The platform makes it easy for teams to very rapidly prototype new application features and functions and present prototypes to users for immediate feedback.	Use case P.
Ease of use	The platform makes it easy for relatively non-technical 'citizen developers' to deliver relatively simple business applications.	Use cases D, P.
Security	The platform offers rock-solid features to secure sensitive business data and prevent unauthorised use.	Support full range of application scenarios (including customer-facing).
Scalability	The platform is quick to work with at first, but is also easy to scale in two ways: to support high-use application deployments, and also to support large numbers of applications.	Support true transformation efforts.

[1] CIOs' responses to digital imperatives highlight gaps in skills and trust, May 2017 -