Arthur D Little PRISM

Unleashing innovation using low code/no code – The age of the citizen developer

Greg Smith, Michael Papadopoulos, Joshua Sanz, Michael Grech, Heather Norris

As the saying goes, "software is eating the world," forcing companies to change their business models and operations. In a highly competitive environment, ensuring they have the required software capabilities is a challenge. This article shows how adopting low-code/no-code techniques enables businesses to widen their options, providing tools for non-specialist "citizen developers" to digitize and transform their operations.

Over the last decade we have seen the "softwarization" of products, services and experiences. Organizations that spent decades developing and improving their business models and core capabilities have been outcompeted by those that disrupted their own business models with software. It has been famously said that "software is eating the world"[1], and in today's highly competitive environment, organizations find it challenging to fulfill the development and operational demands needed to keep their businesses running, while continuing to expand and enhance their digital capabilities.

In this article we will explore how enterprises can leverage today's low-code/no-code (LC/NC) solutions – declarative[2] development options with relatively low learning curves that provide the company's workforce with tools to easily create software to grow and transform the business.

What is low-code/no-code (LC/NC)?

Complex tasks have been progressively simplified through the use of technology and visual tools. For example, word processing has been so affected and simplified by visual programming that today's average seven-year-old can compose a well-formatted electronic document with graphics in minutes.

[1] "Why Software Is Eating the World", Marc Andreessen, The Wall Street Journal, 2011
[2] Declarative programming is a programming paradigm – a style of building the structure and elements of computer programs – that expresses the logic of a computation without describing its control flow.

Low-code and no-code tools are the latest stage in this evolution. They allow anyone to create application software through graphical user interfaces and declarative programming. LC/NC development platforms provide tools for designing applications or systems, along with their required inputs, outputs and business logic. LC/NC takes a visual approach to build business applications more efficiently. This allows "citizen developers" – such as designers and businesspeople – to build and test applications quickly, without the need for developers.

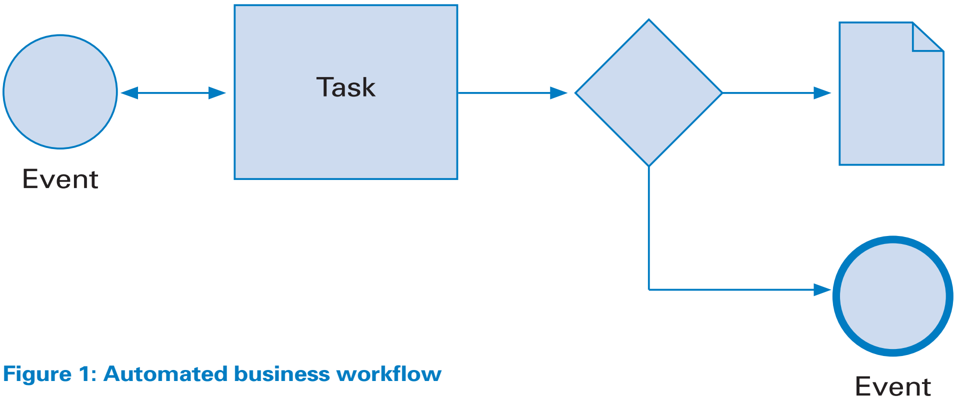


Figure 1: Automated business workflow

Low-code and no-code platforms enable average people to create apps quickly, and then layer customization and added features on top of them. With these options, enterprise users may develop their own apps without writing a line of code, or while keeping code to a minimum for any optional customization. According to Gartner's research[3], "by 2024, low-code application development will be responsible for more than 65 percent of application development activity."

The difference between low-code and no-code: Whereas low-code solutions usually require some initial installation, deployment and minimal coding skills, no-code platforms are faster to set and much easier to use, and hardly require any coding for the most basic functionality.

[3] Magic Quadrant for Enterprise Low-Code Application Platforms–Gartner

No-code platforms are good for building line-of-business apps when you have limited IT resources and technically savvy businesspeople. Low-code development platforms, on the other hand, are a good choice for professional developers who want to create long-term custom applications that will help implement business requirements but have much higher customization needs – such as pixel-perfect user interfaces.

Benefits of LC/NC

For any business undergoing a digital transformation, a LC/NC approach to development can enable a wider range of people and empower the workforce. It builds on innate creativity and entrepreneurship to automate back-office tasks, which can improve internal operations and execute ideas by quickly moving them from inception to prototyping. It allows anyone from the business without previous technical experience to become a smart software generator, which takes the pressure off IT and enables innovation to come from all sides of the business. It also solves real problems that may not otherwise get the attention they deserve by enabling the people closest to them to solve them. Why not devolve processes and capabilities back to their key stakeholders? This will also allow companies to avoid offshoring in some cases, keeping the solution closer to the problem. Furthermore, it will help address a common challenge in software creation: how to communicate requirements to developers.

Whether it is used as an innovation catalyst, to boost productivity, or to address the developer shortage[4], LC/NC's intuitive user experience, low complexity, and adaptability have a double impact. It brings in new software that would not have otherwise existed, and also decreases development time for software that would have otherwise been implemented in a classic manner. This results in faster time to market and significant overall cost reduction (CAPEX and OPEX) when applied properly. For example, low-code services specialist Green Lemon Company worked with a logistics company to create a vehicle-tracking dealer portal that went from design to deployment in four weeks. The estimates using the traditional development indicated a four- to five-month effort[5]. Our own case studies (see Box 1 later in this article) show similar time-to-value gains.

[4] The New Frontier of Software Development–Salesforce research

[5] Digital Enablement via Low Code Platform–TechMarketView

In addition, LC/NC solutions facilitate a user-centric approach with great flexibility to adapt to customer needs and evolve the solution. It skips front-end involvement of IT when starting work on business-critical apps and brings them in when it is time for live integrations, security, deployments and the like.

When should you use LC/NC?

LC/NC solutions are ideal for non-technical staff members who understand the business context and are best positioned to drive and challenge requirements. A solution can start with the business, and then be handed over to IT for enhancements. This eliminates long IT backlogs[6] or having to procure services from external parties if the company does not have in-house IT capabilities. Most benefits of a LC/NC approach come when it is applied to productivity use cases that require strong business knowledge or projects that have limited initial investment and tight deadlines, and require fast turnaround and flexible approaches. Typical applications are as follows:

- **Operational efficiency improvement:** Automation of manual and paper-based processes to increase accuracy and quality while reducing costs. LC/NC platforms make it easy to create applications for capturing information and include ready-made connectors for integrating applications into core systems. Most LC/NC platforms include workflow automation. Workflow tools allow the people closest to the business, such as a business analysts, to build the processes and simple touchpoints needed for common customer and employee activities. The tools have easy-to-use graphical user interfaces to model processes, and some allow APIs and pre-built integrations for a wider range of use cases so workflows are fully integrated with in-house and third-party systems. However, one disadvantage is they can only provide part of the solution and offer limited ability to alter the user experience; hence, further work may be needed for full automation and integration with other systems.

[6] The No-Code Solution to Your Digital Transformation – Betty Blocks

- **Innovation and prototyping:** An often-overlooked capability of LC/NC solutions is innovation and "art of the possible" applications. LC/NC solutions allow business users – the people who best know and understand the challenges and the vision – to unlock value by "smart stitching" emerging technologies such as the Internet of Things (IoT), artificial intelligence (AI), machine learning, and even blockchain together with older, legacy data. This allows for an iterative test and learn-and-discover approach, in which users create functional prototypes and iterate towards fully functional applications.

- **Smart stitching[7]:** The availability of open APIs, microservices, and developments around function-as-a-service (FaaS) and serverless applications underline how new-generation applications are moving towards collections of connected services[8]. Over time, organizations often end up with big suites of tools and systems (bespoke and off-the shelf) from different vendors. This is usually the case for global enterprises that have expanded and diversified through both acquiring other companies and setting up operations in new markets by building applications quickly, without global IT growth strategies in mind. LC/NC platforms enable rapid creation of pinpoint, domain-specific applications, which, via connectors or building middle layers, can be "stitched" together with legacy applications to create end-to-end products. Businesses can use LC/NC tools as tactical solutions to integrate different components to "keep the lights on" while they decide if they need to replace potential burning platforms in the long term.

[7] Smart stitching (http://www.digitalproblemsolving.com/meme/smart-stitching) | Arthur D. Little

[8] Digital Enablement via Low Code Platforms – TechMarketView

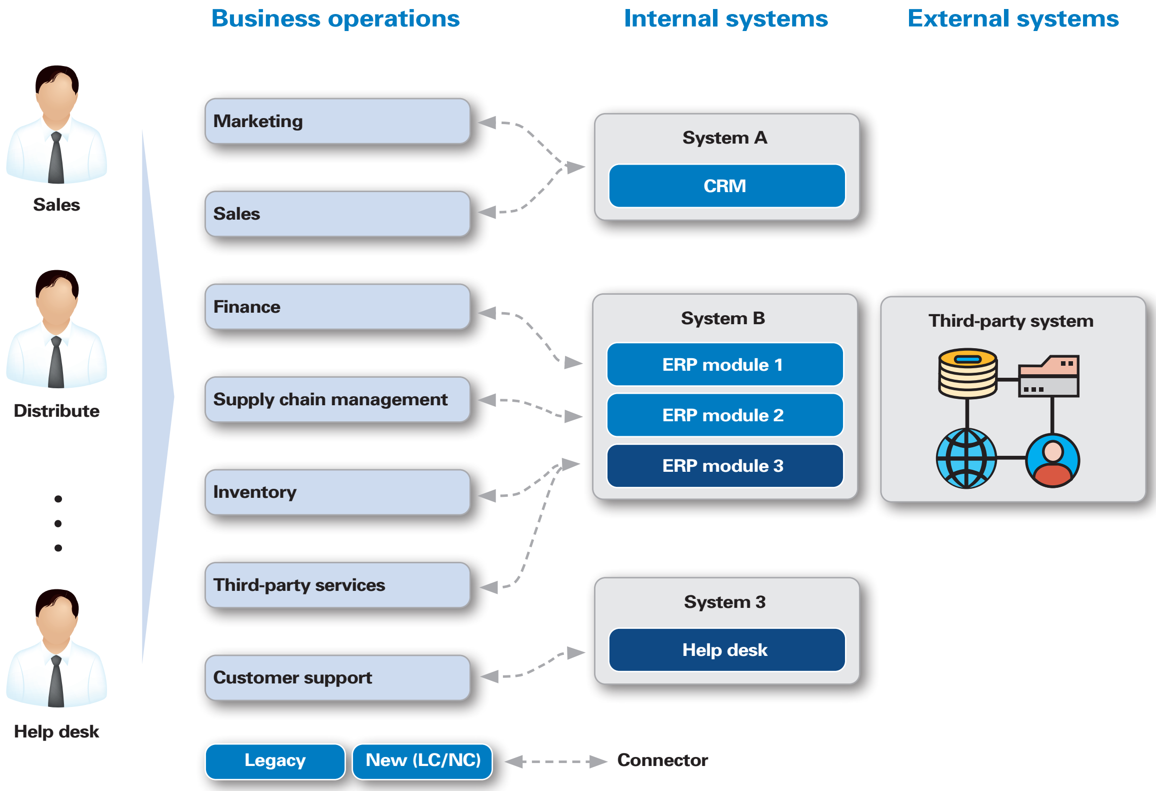


Figure 2: Easy expansion of operation systems with LC/NC tools

In Box 1, we have summarized a few case studies in which ADL has implemented LC/NC solutions to help clients:

Box 1 – LC/NC case studies

Mobile network operator value-added services: ADL used LC/NC tools to design and implement a plethora of value-added services (VASs) for a mobile network operator. These ranged from the standard services of balance enquiries, balance transfers, and credit top-ups to more innovative services such as quiz games. The services were all accessed via SMS or the USSD protocol, so there was no need for a graphical user interface. The platform used the Java Business Process Management (jBPM5) engine, with the input and business logic designed in a workflow graphical editor. Though the technology used in this example has since been significantly superseded, the core patterns and principles have not. Engineers and developers would establish the core API calls and system API functionality, and then hand these over to business- and product people, who would use these as "Lego" blocks to build the necessary functionality. The platform served over 1 million customers, achieving speeds of over 120 transactions a second and proving LC/NC could be made efficient.

A robot-as-a-service (RaaS) platform for a global company: ADL used a modern LC/NC platform to smart-stitch together a number of off-the-shelf SaaS tools to enable CRM (including leads and campaigns), inventory, subscriptions and help desk. We set up a full greenfield RaaS platform for the European entity of a global corporation. The solution allowed for full customization out of the box for building flowcharts and creating logical steps for the process to make it repeatable, productive, and well structured. The solution also had out-of-the-box connectors to a variety of additional services, such as Amazon Connect, a cloud-based contact-center facility that could also be fully customized using simple drag-and-drop tools.

Limitations and barriers of LC/NC

LC/NC solutions offer significant benefits, but, as might be expected, they are not a panacea. There are some limitations to consider:

- **Technical savvy:** Although LC/NC solutions make it possible for non-technical people to build applications, there could still be barriers to entry from a technical point of view. Aspiring users of LC/NC solutions need to at least know how to operate the tool itself, as well as be aware of some "technical" concepts, such as APIs and databases. For the fully customizable LC/NC options, some code will still need to be written, possibly within the LC/NC tool itself – even if they are basic requirements such as bits of Javascript, CSS and simple SQL queries. Employees need to have the basic technical skills to use the tools, or be able to acquire them, potentially with some light support from the IT team.

- **Lock-in:** All LC/NC development platforms result in lock-in, unlike conventional application platforms. For example, you can't take an app created using Google App Maker and transfer it into Microsoft PowerApps without rewriting the entire application. Some platforms, such as Camunda, allow you to port your application to a different cloud, but not generally to a different tool. Note that vendor lock-in does not have the same perfidious risk of the old days, when there was complete dependency and prices were constantly increasing. Healthy competition between vendors helps drive both innovative new features and functionality on LC/NC platforms, while keeping prices down (and, indeed, reducing year on year). In our view, this is an acceptable risk as long as the competitive differentiation enabled by the app is low.

- **Constrained performance:** LC/NC development tools decrease the amount of time needed to build an application. However, the applications that result from LC/NC development are rarely optimized for performance and efficiency due to the abstract, declarative nature of LC/NC solutions. Simply put, if you need an app that runs as quickly as possible and consumes the fewest resources, you will have to write bespoke code and go as close to the silicone as you can. However, this does not mean "low" or sluggish performance – this constrained performance refers to low parallelism, rather than an unpleasant experience. Most day-to-day users won't care or even notice that the platform showing a well-designed and well-thought-out business dashboard is consuming, for example, 10 percent more memory and 5 percent more CPU power than a non-LC/NC equivalent. LC/NC platforms will thus do very well, and modern platforms can scale to millions of users with reasonable requirements if executed correctly.

Introducing LC/NC solutions into the organization

Companies looking to implement LC/NC solutions should consider a number of important priorities to ensure success.

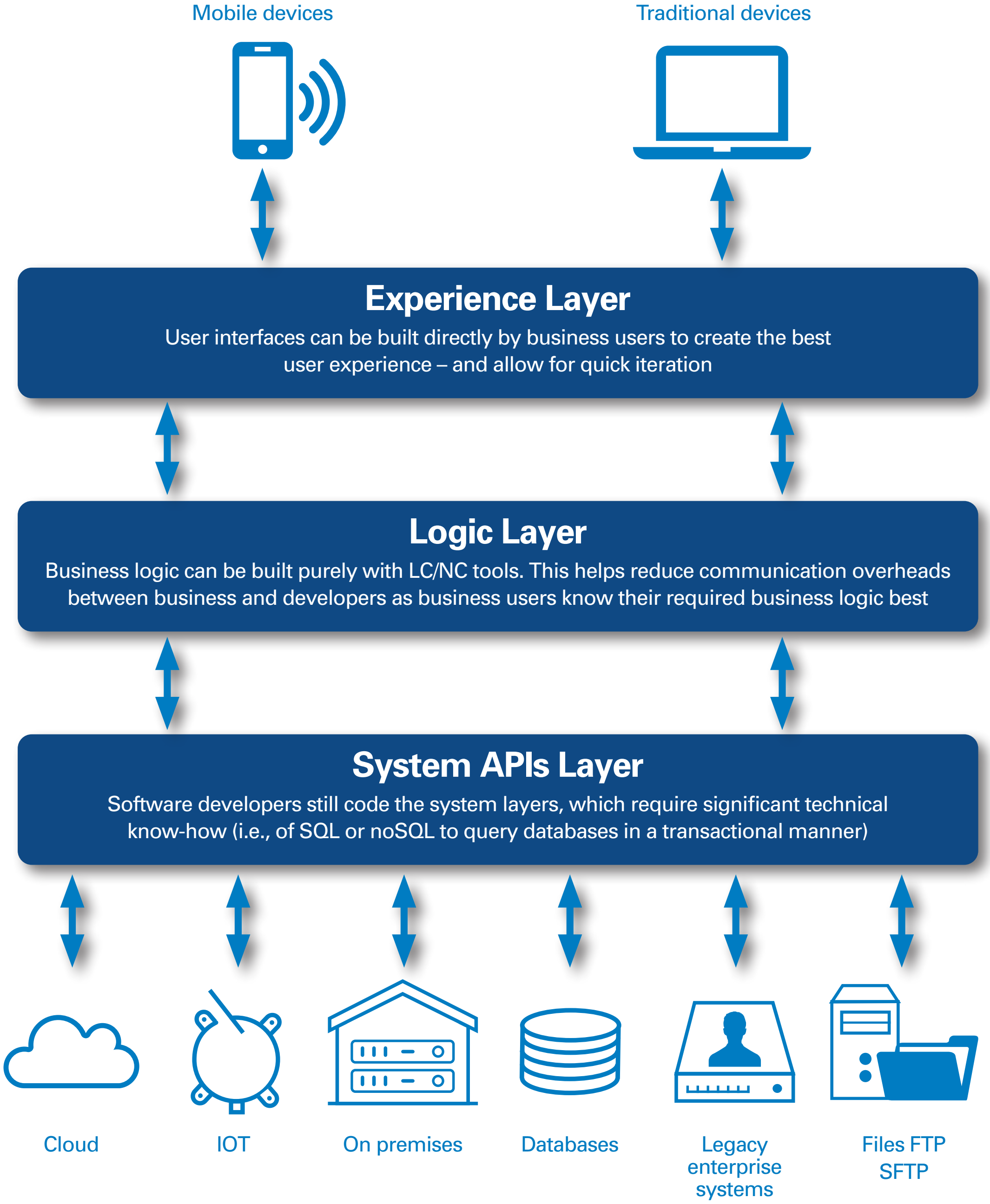
1. Establish business value-add and link to strategic objectives

Before embarking on any implementation, it is important to understand the value that the LC/NC application will deliver to your business and how it links to your strategic objectives. Spending time in a discovery phase will not only validate whether an LC/NC solution is needed in the first place, but also contribute to a shorter development phase later on, which will lead to savings across the life cycle. Then you will need to consider the type of application that best fits with your organization, and finally the organizational changes you will need to implement for its adoption to succeed. This is not just about having the right tool, but also knowing how to engage stakeholders and streamline processes, roles and responsibilities for them to contribute.

Determining how LC/NC tools can best serve your strategic objectives will depend on your organization's capabilities and enterprise architecture. Traditionally, there has been tension between IT and business departments. The former is incentivized to retain tight control over the application landscape to ensure stability and maintainability (which often leads to a rise in "shadow IT"), while the latter prioritizes speed of development and cost control.

2. Put the right system architecture in place

The appropriate architecture and organizational structure can help facilitate the introduction of LC/NC solutions and relieve the tension between IT and business departments. The introduction of a tiered architecture can help separate the technical and non-technical aspects of a system. IT users can then focus on the technical details of the application, while others make use of LC/NC tools that focus on business logic. LC/NC tools can make use of the functionality offered by technical components of the system without being concerned with how they were implemented. Figure 3 highlights how this can be achieved with a three-tier architecture:



Experience Layer

User interfaces can be built directly by business users to create the best user experience – and allow for quick iteration

Logic Layer

Business logic can be built purely with LC/NC tools. This helps reduce communication overheads between business and developers as business users know their required business logic best

System APIs Layer

Software developers still code the system layers, which require significant technical know-how (i.e., of SQL or noSQL to query databases in a transactional manner)

Figure 3: Tiered architecture that enables citizen developers

The system API layer will hide the complexity away from users further upstream. Developers and technical resources with intimate knowledge of databases, servers and other important components will be responsible for this layer. The team responsible for building and maintaining the system API layer will also have the tools necessary to safeguard critical systems from improper use of applications further upstream.

The Logic Layer will make use of the functionality offered by System APIs without being concerned with complex concepts such as database queries and cache management. Business users could plug into system APIs with LC/NC tools to rapidly create new applications based on their knowledge of user requirements. New insight and value can be created just by combining existing data and capabilities. The Logic Layer is where LC/NC tools shine because they allow for quick creation of complex business logic by the people who know it best. The Logic Layer is effectively protected, as the system API – created by technical people – provides the guardrails for performance and security.

Last, but not least, the Experience Layer allows users to create the UIs/UXs they need.

3. Use good change management practice

Statistics show that between 60 and 85 percent of digital transformation projects fail due to lack of good change management practice and technical skills to execute[9]. Researchers have highlighted how critical change management is to driving a successful outcome. They stress the importance of communication and prioritization, especially during the set-up and pilot phase. Simplifying development to offer productivity gains and speed of delivery will require introducing corporate development to non-IT roles, which can be a significant change and cultural hurdle. Without the right training and governance structure, IT will be quickly overwhelmed with product quality and management challenges, and maintenance will become unsustainable. Tools should enable incremental learning, but it is also important to conduct training sessions to teach and enforce good practices. An organization might need to update some of its development guidelines to accommodate non-IT roles within the software development life cycle and make the business accountable for the quality and success of the solutions implemented. Collaboration between business and IT is key – CIOs should act as enablers and champions of change by consolidating communication tools, piloting and encouraging new ways of working, and streamlining processes to foster closer collaboration.

9. Successful Digital Transformation. How Change Management helps you to hold course – Siemens

4. Use minimum standards initially

When working with LC/NC platforms for the first time, the organization's focus should be on meeting minimum operational process and quality standards early on. A faster pace will follow as applications shift from traditional development approaches, in which IT is fully responsible, to a model in which the business is at the center and realizing objectives is in the hands of the people on the operational front line, who have better understanding of business functions.

5. Adopt an agile development approach

If an IT capability already exists in the organization, the introduction of LC/NC does not need to disrupt the delivery framework. As an example, in an agile scrum, the principles, processes and ceremonies will still be followed, but benefits will be maximized by combining the fast decision-making of agile with the fast development speeds of LC/NC. Small adjustments will need to be made to adapt and improve the processes that come from this new way of developing, in which demands on non-IT stakeholders will increase, requiring collaboration to be tighter and more frequent. The same quality assurance rigor that is applied to traditional development should still apply. Apps need to follow the mandatory development life cycle to include testing, security checks and controlled deployments before going live.

Insight for the executive

We don't believe the skill of writing code is becoming obsolete or software developers will lose their jobs anytime soon – with low-code tools you can do something, but some skills and technical knowledge are still needed to understand how you should do it. We do believe that LC/NC platforms are finally achieving their purpose, giving citizen developers the power and simplicity to build software. They save time and money, while accelerating the pace of digital innovation and transformation within organizations.

One of the biggest strengths of LC/NC platforms is that they allow us to take advantage of the innate problem-solving skills of human beings by removing at least a significant portion, if not all, of the barriers to implementing software solutions in today's softwarized world. Simply put, LC/NC enhances the potential of innovation. This should resonate with organizations that have realized the necessity to become software driven if they are to survive, and even thrive, in the coming years. Ultimately, the benefits of LC/NC platforms are crystallized in their ability to empower entrepreneurial people in the organization to unleash their visions, ideas and creativity with minimal support through building smart software solutions.

Greg Smith
is a Partner in Arthur D. Little's London office and a member of the Digital Problem Solving Practice.

Michael Papadopoulos
is a Chief Architect in Arthur D. Little's London office and a member of the Digital Problem Solving Practice.

Joshua Sanz
is a Manager in Arthur D. Little's London office and a member of the Digital Problem Solving Practice.

Michael Grech
is a Manager in Arthur D. Little's London office and a member of the Digital Problem Solving Practice.

Heather Norris
is a Consultant in Arthur D. Little's London office and a member of the Digital Problem Solving Practice.