Aug 5 2020

Low-Code/No-Code Automation Enterprise

What's new & interesting in low-code/no-code

- Lately, it seems like everyone is talking about low-code/no-code platforms. Analysts are forecasting that 65% of enterprise apps will be developed on low-code platforms by 2024 (up from 10-20%).
- The advances of the past decade lie at the foundation of what makes the current wave different – responsive design, cloud, automation, APIs and "API-first" companies.
- An incredible array of low-code and no-code players have emerged – ranging from big tech giants to niche startups, integration players to high-design website creators, consumer app platforms to enterprise app tools.
- At some point in each low-code/no-code hype cycle comes the realization of the technology's limitations. Engineers will not become obsolete – the more likely scenario is we'll see even more demand for them. They may, however, shift emphasis towards a different problem set.
- This is the "gold rush" of the low-code/no-code movement. The removal of technical barriers to app development is bringing many more ideas to market for validation. People will be able to focus on where they can offer value, whether it's content or community or brand.

"Low-code/no-code is accelerating application development, and allowing users and developers to focus on the unsolved specifics of each problem."

What It Means section

Brief Guide

Use quick links to jump to a section.

What's Happening

Lately, it seems like everyone is talking about low-code/no-code platforms, from Webflow to Notion to the recently announced Amazon Honeycode. IT and developers are facing overwhelming demand, putting pressure on enterprises to seek alternatives. **The COVID-19 crisis is also accelerating the boom in low-code/no-code, which is helping government agencies and businesses build apps quickly to serve new needs.** In some cases, low-code/no-code can reduce development time by as much as 90%. One bank, for instance, built an app to track the influx of PPP loan applications within 5 hours.

Low-code/no-code startups are on track to raise $500M+ in venture capital this year, at multi-billion dollar valuations in some cases. Gartner is forecasting that **65% of enterprise apps will be developed on low-code platforms by 2024 (up from 10-20% in 2019)**. Other industry analysts are projecting a low-code market worth $14B to $53B within 4-6 years.

Low-code/no-code development isn't exactly new. It has existed in various forms for decades, going back to the CASE (computer-aided software engineering) tools of the '80s and '90s, followed by Microsoft Excel VBA, the original Dreamweaver, WordPress, the app template platforms and Jira workflow engine of the early 2010's, and the more recent wave of modern low-code/no-code platforms.

As the name suggests, low-code/no-code software development seeks to alter the paradigm of traditional language-based programming (e.g. C++, Java, Python) for application development. Low-code/no-code platforms use more intuitive, structured interfaces, workflows, logic and data to make it easier and faster to create new applications. **These mechanisms might include drag-and-drop blocks, modules, templates, plugins, themes, widgets, connectors, menus, "if this, then that" chains, natural-language inputs, and domain-specific languages.**

So what's new about today's low-code/no-code platforms?

Today's low-code/no-code platforms build upon the advances of the past decade – such as responsive design, scalable cloud environments, cloud-based services and applications, modern devops, automation technologies, and perhaps most of all, **APIs and**

"API-first" companies. These advances lie at the foundation of what makes the current wave different.

The **capabilities offered – which vary depending on the platform – include**:

- **More complex, full-featured web-based and mobile applications** (e.g. user signups, payments, search, analytics)
- **Modern user experiences** (e.g. polished, responsive, cross-platform)
- A **vast array of low-code/no-code integrations** and **automations of extended workflows** across disparate 3rd-party platforms
- **Offerings from big tech giants integrated with cloud services** (e.g. Microsoft Power Apps on Azure, Google's AppSheet, Amazon Honeycode) **and collaboration tools**
- Built-in "escape hatches" that allow for **code extensibility/customization** (even among "no-code" players) and better-quality **code export** (so users aren't locked into the platform)
- Cloud-based platforms that benefit from **continuous updates and improvements**
- **Better reliability, security and compliance controls** to serve mission-critical enterprise applications
- **Tools built for both developers and end-users** – such as flexible deployment options (e.g. cloud or on-premise) and intuitive drag-and-drop tools
- **Domain-specific platforms for functions and industries** (e.g. HR, financial services)

The flood of low-code and no-code players

It would be hard to fully cover the incredible array of low-code and no-code players, many of which have emerged within the past couple years, with more continuing to pop up every day. They range from **big tech giants** (such as Microsoft Power Apps) to **niche startups** (such as Glide), **integration players** (such as Zapier) to **high-design website creators**

(such as Webflow), **consumer app platforms** (such as Bubble.io) to **enterprise app tools** (such as Salesforce Lightning).

The distinction between low-code and no-code is also blurred. Both use intuitive interface elements (such as drag-and-drop) to generate code. **In theory, low-code platforms are also extensible with custom code, while no-code platforms rely on the prebuilt interface elements**. In practice, however, most low-code platforms can be used without custom code, and most no-code platforms have "escape hatches" that allow for custom code.

The reality is that **"no-code" has become a marketing handle used to position companies in a hot space. All these platforms use code, most generate code, and some allow for extensibility with custom code.** Webflow, for instance, is extensible and technically low-code but positions itself as a no-code champion with a conference called the No Code Conf. Others like Betty Blocks also call themselves "no-code" but allow for custom code.

The difference between "no-code" and "low-code" is often based on software generation and target audience. More established platforms (e.g Zoho Creator, Appian) targeting enterprises and developers are more likely to use the term "low-code." In contrast, newer platforms targeting designers and less technical end-users tend to use "no-code."

This is further complicated by **these terms increasingly being used in reference to what would previously have been known just as "software."** For instance, a subscription newsletter platform like Substack might now be called "no-code" because it brings together elements that once required coding, such as subscriptions, payments, and community features. In general, **being a "low-code/no-code" player implies a measure of design flexibility and ease of use in building functional software that would previously have involved developers.**

Our goal with the sections below is to present a slightly more orderly view into a chaotic market environment, highlight the most notable players, and offer a sense of the richness of the ecosystem, without being fully comprehensive.

Integrations & workflow automation

Low-code/no-code players have arisen to help businesses and enterprises integrate across internal and 3rd-party resources (e.g. applications, data sources), and build custom

workflow automations. **These services typically rely on prebuilt APIs and integrations, or offer low-code tools to create APIs**.

- **Zapier** – A well-known workflow automation platform with prebuilt integrations across 2,000+ 3rd-party applications, Zapier puts a strong emphasis on business and productivity (e.g. Google Sheets, Slack, Gmail). It has 100K+ paying customers, 5M+ users and $50M+ in revenue.

- **IFTTT** (If This, Then That) – One of the earliest platforms to take a trigger-action recipe-based approach, IFTTT is popular for smart-home automation (though it is also used for business use cases). It has 20M+ users, 650+ integrations, and 1B+ applets run every month.

- **Integromat** – An "online automation platform" with advanced logic, Integromat has integrations with popular apps and services as well as HTTP/SOAP and JSON modules that enable no-code integration to most web services.

- **Other enterprise integration platform-as-a-service (iPaaS)**: These cloud-based integration platforms for larger enterprises include Dell's **Boomi**, **Workato** ($70M raised in Nov 2019), **Tray.io** ($50M raised in Nov 2019), Celigo's **Integrator.io** ($20M raised in Oct 2019), **SnapLogic Intelligent Integration Platform**, **Jitterbit**, **Zoho Flow**, **Automate.io**, **Blendr.io**, Salesforce/Mulesoft's **Anypoint Platform**, **IBM App Connect**/**IBM Automation Platform**, **Informatica**, **SAP Cloud Platform Integration Suite**, **Microsoft Azure Integration Services** (including low-code Azure Logic Apps; see big tech section below), and **Oracle Data Integration Platform.**

- **Other tools/platforms**: e.g. HubSpot's **PieSync** (two-way customer data synchronization across applications), **Unito** (two-way synchronization across productivity tools), **Jira Cloud** (no-code workflow automation across Jira products); **Tonkean** (no-code workflow automation; $24M raised in Apr 2020), **JIFFY.ai** (enterprise automation platform; $18M raised in Jun 2020), **Process Street** (no-code workflow builder; $12M raised in Feb 2020), **Anvil** (automates paper-intensive workflows; $5M raised in Jun 2020), **n8n** ("fair code" workflow automation), **Node-RED** (flow-based visual programming tool, originally designed for wiring together IoT), **Pipedream** (low-code integration platform for developers), **Hevo** (no-code data-pipeline integrations).

Low-code/no-code application development: Front-end & ecommerce

Low-code/no-code for front-end, consumer-facing applications (e.g. websites, apps) has evolved to offer **more beautiful and interactive experiences out of the box**:

- **Webflow** ($72M raised in Aug 2019) – Webflow is popular amongst designers and business users for its ability to enable **modern responsive websites**, integrate services such as content management or ecommerce, and support customization or code export if needed.

- **Shopify** (Q2 2020: Revenue up 97% to $714M, with $30B in gross merchandise volume) – Used by 1M+ businesses globally to operate and market their offerings online, Shopify's platform supports **ecommerce websites that are deeply integrated with its suite of tools** (e.g. shopping cart, payments, store management) and integrations (e.g. fulfillment, social media).

- **Other website platforms:** e.g. **Squarespace** (grid-based platform that was the gold standard for small businesses before Webflow), **Wix** (no-code beginner-friendly platform for smaller ecommerce businesses), **Wordpress** (still widely used, though older, content management platform with extensive ecosystem of plugins and templates).

- **Mobile, desktop & web apps (general):** e.g. **Bubble** ($6.3M+ raised in Jun 2019; full-featured responsive web apps with native wrappers), **AppGyver** (full-featured native mobile & web apps), **Adalo** (no-code native mobile apps), **Thunkable** (no-code native mobile apps), **GoodBarber** (native & progressive web apps), **Draftbit** (React Native cross-platform apps), **Buildfire** (cross-platform mobile apps).

- **Landing pages, forms & surveys:** e.g. **Carrd** (responsive one-page websites with integrations), **ConvertKit** (conversion-focused landing pages, forms & email marketing), **Unbounce** (conversion-focused landing pages), **Instapage** (unique landing pages for each ad), **Typeform** (embedded forms & surveys designed as a conversation), **JotForm** (form builder)

- **Voice, chatbots & other brand experiences**: e.g. **Voiceflow**, **Vuix** and **ApiToBot** (voice apps for AI assistants); **ManyChat**, **Engati**, **Joonbot**, **Landbot**, and Kustomer's **Reply.ai** (chatbots), **Scapic** (augmented-reality shopping), **Ceros** (interactive brand experiences).

Low-code/no-code application development: Big tech firms

Big tech firms have been doubling down on low-code/no-code, with a spate of recent activity:

- **Microsoft**, which expects 500M new apps to be built in the next 5 years, has been **actively investing in low-code for its Power Platform suite** of business tools. In addition to the **Power Automate** workflow automation platform (formerly Microsoft Flow), Microsoft launched **Power Virtual Agents**, a no-code chatbot platform, in Nov 2019. It followed in Apr 2020 with the **no-code beta version of its Power Apps (Microsoft's fastest-growing business application ever)**. Last week, it announced the new **Dataflex for Teams**, a low-code data platform built into Teams. Separate from the Power Platform, Microsoft launched **Lists** for Microsoft 365 users – an Airtable-like tool with templates and deep integrations with Microsoft's other applications. (On the cloud side, Microsoft also offers **Azure Logic Apps** with out-of-the-box connectors for no-code workflow automation.)

- **Amazon:** In late Jun 2020, AWS announced the beta launch of **Amazon Honeycode, a managed service for users to build custom low-code/no-code web and mobile applications**. Honeycode uses spreadsheet-style databases for storing application data, drag-and-drop design for interfaces and capabilities, and actions for workflow automations. Honeycode, which is offered as an AWS service, is free for up to 20 users and 2500 rows of data per workbook. In conjunction, Amazon is offering app-building courses, a knowledge center, and community forums. Honeycode was available in one AWS region at launch, and AWS customers Slack and SmugMug have already said they plan to use the service.

- **Google:** In Jan 2020, Google announced it was shutting down its low-code App Maker platform (launched in 2016) in 2021 due to low usage. At the same time, it announced it had acquired **no-code app development platform AppSheet**. Housed under Google Cloud, **AppSheet will be integrated with G Suite** and allow users to develop applications that leverage both Google data sources (e.g. Sheets, Forms, Android, Maps, Google Analytics) as well as 3rd-party sources such as Salesforce, Dropbox, AWS DynamoDB and MySQL. (AppSheet already has geolocation and machine learning services built in.)

- **Apple:** In mid-2019, Apple **rebranded its longstanding FileMaker platform for building low-code apps to Claris**. At the same time, it announced the **Claris Connect** service (based on its Stamplay acquisition), which will help users connect

APIs and automate workflows using low-code tools. Claris is investing in brand awareness, workflow applications, and new types of applications (e.g. IoT).

Low-code/no-code application development: Other enterprise platforms

This category includes some of the **more established, larger platforms** previously known as "Enterprise High-Productivity Application Platforms-as-a-Service" or HPaPaaS (Gartner). The category has been referred to more recently as Enterprise Low-Code Application Platforms (LCAP) and Multiexperience Development Platforms (MXDP). Mendix, Outsystems and Salesforce are the three vendors that have landed in Gartner's top-right quadrant for both LCAP and MXDP.

- **Mendix**, acquired by Siemens in Aug 2018 for $700M, offers a low-code, cloud-native platform to "design, build, integrate, test, deploy and manage enterprise apps at scale." In addition to its **AI-assisted visual development environment for multi-channel apps** (web, mobile, conversational, AR/VR), Mendix emphasizes **business-IT collaboration**, **governance/control** across the application development process, and **flexible deployment**. Mendix has strategic alliances with SAP, IBM and Pivotal, and also supports Siemen's Xcelerator.

- **Outsystems** offers an **intuitive low-code visual development platform to create full-stack cross-platform apps with an emphasis on speed of development**. The open, standards-based platform includes features such as single-click deployment, automatic refactoring, and offline data synchronization. Outsystems recently launched an enhanced partner program with certification paths, industry and use case templates, and licensing programs to help partners such as Deloitte and KPMG bring new offerings to market.

- **Salesforce** has its own **low-code/no-code platform called Lightning,** which can create applications from spreadsheets, build workflow-based applications, and develop custom apps that draw from other Salesforce products. Salesforce also rolled out new AI-powered no-code data integration and workflow automation features for its Mulesoft Anypoint Platform in Nov 2019.

- **Other low-code application platforms:** e.g. **Betty Blocks**, **Appian**, **ServiceNow Now Platform**, **Oracle APEX**, **Kony**, **Pega**, **Zoho Creator**, **Agilepoint**, **TrackVia**, **Quick Base**, **Kintone**, **Nintex Workflow Cloud**, **Caspio**, **WaveMaker**, Zudy's **Vinyl**, **AuraPortal**.

Low-code/no-code application development: Vertical platforms

- **Unqork** ($131M Series B finalized in Feb 2020) focuses primarily on **insurance, financial services, and public-sector use cases**. It has seen massive growth during the COVID-19 crisis, with Q1 2020 revenue up 320% year-over-year. Unqork partnered with Deloitte to help small businesses procure CARES Act loans, and recently followed with a strategic alliance to deliver solutions in financial services. It also partnered with KPMG to offer a mortgage forbearance /loss mitigation tool, and with DC and NYC city governments to develop support hubs for residents.

- **Bryter** offers no-code automation for enterprise use cases in support functions such as **accounting, legal, compliance, and marketing**. Bryter, which raised $16M in Jun 2020, has 50+ enterprise clients including Deloitte, PwC, KPMG, McDonald's, Telefónica, ING, Hogan Lovells, and Baker McKenzie.

- **Other vertical tools:** e.g. **Sharetribe** (marketplaces), **Substack** (newsletters), **Palabra** (automated emails), **Oyku** (podcasts & YouTube shows), **Buildbox** (games), **Sora** (HR), **Wandelbots** (industrial robots; $30M raised in Jun 2020), **INSTANDA** (insurance), **Appway** (financial services), **Enduvo** (AR/VR), **CloudWorx** (IoT), **Turbo Systems** (engagement apps), **Modulz** (design; $4.2M raised in Sep 2019).

Low-code/no-code application development: Based on spreadsheets & databases

- **Airtable** – "**Part spreadsheet, part database**," Airtable allows non-developers to create flexible databases that can be used as a back-end (e.g. for websites, apps, and workflows) but manipulated like a spreadsheet. Airtable was last valued at $1.1B in 2018, and was reported in Apr 2020 to be looking to raise $50M at a valuation of $2.5B to $4B.

- **Other tools**: e.g. **Parabola** (drag-and-drop platform for repeatable data tasks that would otherwise be done on a spreadsheet); **Glide**, Google's **AppSheet**, **Sheet2Site**, **Pory.io**, **Siteoly** (formerly NoCodeSheets), and **Open as App** (apps/websites from Google Sheets, Excel, Airtable or other spreadsheets/databases); **Dashdash**, **Actiondesk**, and **Better Sheets** (high-functioning spreadsheets with integrations); **Knack** (cloud database for apps).

Low-code/no-code application development: Based on documents & digital workspaces

- **Notion** offers a **lightweight "all-in-one workspace"** (e.g. Notes & Docs, Knowledge Base, Tasks & Projects, and Spreadsheets & Databases) that can also serve as a website, web app, desktop app or other internal tool. As of early Apr 2020, when it raised $50M at a $2B valuation, Notion had 4M+ users and annualized revenue of $30M. It is extending its platform to bring on larger customers (e.g. permissions, developer API).

- **Coda** uses **high-functioning "programmable documents"** that bring together traditional documents, spreadsheets, and other forms of media, and can be turned into an app or website. Coda integrates with 3rd-party apps like Slack, G Suite, GitHub and Figma, with the ability to bring data in or push data out through Coda's "Packs."

- **Other tools:** e.g. **Kissflow** (digital workspace for automated internal workflows), **Snapboard** (dashboards for internal apps & tools).

Low-code/no-code application development: The long tail

- **AI & data science:** e.g. **Gyana** (data science analytics on structured data), **MonkeyLearn** (AI text analysis), **Obviously.ai** (data predictions using machine learning), **Apteo** (no-code analytics), **Metaranx** (no-code AI platform), **SnapLogic Data Science** (drag-and-drop machine learning), **Informatica Data Engineering Streaming** (low-code IoT and stream processing).

- **Developer tools**: e.g. **Hasura** (point-and-click database relationships), **Kumologica** and **Refinery** (low-code serverless integrations), **Codota/TabNine** and **Kite** (code auto-complete), **Generato** (more efficient custom development), **Retool** (faster development of internal tools), **Crowdbotics** (full-stack app builder), **8base** (low-code backend-as-a-service), **Hyperfiddle** (database applications), **Reflect** (automated testing of websites & web apps), **Waldo** (functional/UI mobile testing), **Autify** (web-app testing).

- **Platform add-ons:** e.g. **NoCodelytics** (no-code analytics for Webflow), **Memberstack** (no-code logins and payments for subscription offerings on

Webflow), **Trint** and **Temi** (automated audio-to-text transcriptions integrated with Zapier).

Supporting ecosystem

A key aspect of what is interesting about what's happening in low-code/no-code right now is the vibrant and thriving surrounding ecosystem – from communities with tutorials and templates, to no-code consultancies and mentors, to podcasts and newsletters:

- **Communities: Makerpad** (no-code community & education), **Twitter #nocode** (Twitter community), **Webflow Community** (70,000+ members), **Nucode** (no-code maker community of 2900+), **No Code Founders** (community of 2000+ founders), **NoCodeDevs** (global community of 750+)
- **Other education & resources: Zeroqode** (low-code/no-code templates), **Automate All The Things** (education), **NoCodery** (education & job board), **Webflow University** (education), **Nocodify** (Bubble.io education), **Nocode HQ** (templates & tutorials), **Codeless.How** (education) **Doc Williams' Build with Me** (YouTube channel), Nelson Abalos Jr's **Pixel Geek** (YouTube channel)
- **Directories: No Code List**, **NoCode.Tech**, **Nocode Essentials**
- **Consultancies/agencies: Edgar Allen** (no-code brand agency), **No Code Mentors** (mentor-maker matching), **ELab** (no-code app development), **Optemization** (startup consulting)
- **Podcasts: Visual Developers Podcast**, **CodeLess: A NoCode Narrative**, **No Code No Problem**, **No Code Podcast**, Bubble's **The No-Code Hustle**, Webflow's **Coffee Talks**
- **Newsletters: Inside NoCode**, **No-Code Coffee**, **No-Code Report**, **No-Code Life**, **NoCode**

Challenges & events: 100DaysofNoCode (no-code an hour per day for 100 days)**,** Webflow's **No Code Conf** (conference), **Nocode Rumble** (challenge)

What It Means

The low-code/no-code movement has both its evangelists and its naysayers. **The history of software development has been an uneven journey towards higher levels of abstraction**. These higher levels of abstraction hold the promise of more efficient reuse of code and faster development cycles. When the abstractions are intuitive (such as drag-and-drop interfaces), the development tools become more accessible to end-users, who may be less technical but are closer to the problem set.

One of the key advantages of low-code/no-code is that it lets users stand on the shoulders of the developers who have already solved the lower-level problems. **It means that users – both end-users and developers – can focus on the creative and cognitive work of solving the specifics of the unaddressed problem set.** They can spend time, for instance, on designing, testing, and iterating upon and enriching the experience. In general, we are seeing apps and websites become more like spreadsheets – a readily manipulable and useful abstraction.

Drivers of the low-code/no-code wave in the enterprise

If Microsoft is right and there's demand for 500M new apps in the next 5 years, **we're facing a shortfall of 1M developers.** In the US, constraints on employment visas could aggravate the shortage even further. **The relative scarcity and cost of developers has been a longstanding bottleneck on innovation**, both inside enterprises as well as among startups. **Low-code/no-code tackles this challenge from multiple angles** – by allowing less technical users to build some of their own apps, by letting users prototype and test apps before custom development, and by streamlining integration and app development/customization for developers.

Businesses facing remote workforces and budget constraints during the COVID-19 crisis are also seeking to gain visibility into operations and automate business processes, as well as serve new needs. At the same time, IT departments – with limited bandwidth and working remotely – are triaging projects. Executives are looking to low-code/no-code to move projects forward and accelerate prototyping and development. One Forrester report

suggests that low-code/no-code can create some software **up to 10x faster than traditional programming**.

The limitations of low-code/no-code

However, abstractions can limit as well as empower – they can put up barriers that prevent users from having a fine degree of control. **At some point during each low-code/no-code hype cycle comes the realization of the technology's limitations** – when a subset of developers and business users hit the wall of what is possible and, in some cases, need to rebuild custom applications to meet their needs. (The irony is that the more successful the application, the more painful it can be to rebuild.) This pullback – the trough of disillusionment – is where low-code/no-code in the past got the somewhat unjustified reputation of being only good for "toy apps."

We're currently in the heat of the hype cycle and will likely hit another trough of disillusionment within the next few years. History doesn't repeat itself but it often rhymes. However, that doesn't mean there's not enormous value in the current low-code/no-code wave. It just means that there are inherent tradeoffs, particularly when it comes to no-code.

No-code is more "opinionated" than low-code, serving well-bounded use cases. **If the goal is to, in Alan Kay's words, "make simple things simple and complex things possible," no-code can do a beautiful job with the former but has more failure modes with the latter.** Low-code, in contrast, offers a "trapdoor" or "escape hatch" when users reach the limits of the platform, which means greater flexibility and more use cases that can be served (though these escape hatches can also land users into the Turing tarpit, "in which everything is possible but nothing of interest is easy.")

The extensibility of low-code platforms also offers more avenues for ecosystem-driven innovation, which means **low-code platforms have the potential to evolve faster**. While no-code has become the term du jour – an easier handle for marketing and a more appealing gateway for non-technical users – **the direction of the market is towards low-code.** Even if it's called no-code.

How low-code/no-code will change IT

We're not headed towards a world where engineers will be obsolete. They may, however, shift emphasis towards a slightly different set of problems – less well-understood problems with many edge cases, harder problems at the next level of complexity, architecture and vendor optimization, data-pipeline engineering, niche problems not well-addressed by existing no-code solutions, and new-frontier problems (e.g. quantum applications). **The more likely scenario is that we'll see even more demand for engineers** due to the Jevons paradox – when resources become more efficient, we tend to use them more.

Developers will also need expertise in low-code/no-code – among other reasons, to address "the last 3% gap" in needs that are unsolved by existing tools. There's the **potential for businesses to fundamentally change how they think about in-house software development.** All IT departments have constraints, and many are facing delays on the product roadmap and a deep backlog. In one Salesforce survey, 72% of IT leaders said backlogs prevented them from working on strategic projects. Building effective low-code/no-code capability can lighten the load and also support agility as a competitive advantage – freeing up time for IT to work on strategic projects that touch the customer. In the longer run, just like the cloud, low-code/no-code will eventually become table stakes.

IT leaders may need to deconstruct the work of IT based on how projects can be addressed with low-code/no-code tools, and how responsibilities (e.g. coding vs. non-coding) are distributed among IT staff. This can help optimize the use of IT's time, with **low-code/no-code serving as a force multiplier.** Similar to serverless computing, **reusable blocks of code are being glued together** – except that rather than having engineers gluing services together with code, less technical users are using the low-code platform's visual programming mechanisms. End-users or "citizen developers" can do much of this upfront work and then **allow professional developers to take it across the finish line** (with customization where necessary).

If there are a thousand flowers blooming in an enterprise, then **governance, control and security of these applications will necessarily become more prominent among IT's responsibilities**. There will be opportunities to recapture "shadow IT" and bring it under the IT governance umbrella, facilitated by enterprise-friendly vendors. Depending on the product strategy, low-code applications may still need to go through planning, version control, testing, deployment, monitoring and documentation.

Among the considerations for IT planning is the possible need to rebuild low-code/no-code applications (which can be "patchwork") **as they scale.** An "exit strategy" should be considered during the earlier phases. No-code features can be a form of technical debt if apps need to be later rebuilt. Pushing the limits of a low-code platform can also result in hacks that generate complexity. Part of the answer is **good vendor selection to avoid lock-in and reduce transition friction** (e.g. ability to export clean code). Applications built on some proprietary platforms, for instance, may not transition well or readily fit into existing devops workflows.

Low-code/no-code imagination and the ecosystem

The momentum behind the low-code/no-code movement is being driven by the broadening of the public imagination around the use cases – what problems could be solved today. IFTTT and Zapier have been around for a while, and despite their substantial growth, the limiting factor has been the number of people who understand the power of these prebuilt integrations and apply creative energy to imagining what could be accomplished. It's a mark of maturity of where we are in this current technological cycle that many problems are well enough understood to create templates for highly useful low-code/no-code applications.

The thriving ecosystem underway today – the communities, mentors, learning platforms, template libraries, and so forth – is what will make the movement last. While some of the vendors will be consolidated, driven into niches, or washed away over time, there will be many more people who understand and are engaged in application development when the tide goes out.

Among the winners will be scalable, standards-based platforms, with robust support and maintenance as well as extensive integrations and partner ecosystems. Enterprises tend to go with the market leaders, since they don't want to end up with 16,000 apps dependent on an obsolete platform. Developers are also looking to trust their tooling and reduce their dependencies, as well as manage version control and automate deployment. **All in all, we can expect the big to get bigger.** However, they won't get to all the niches – at least for some time. There will still be room for specialized and vertical platforms, and tools targeted to micro-communities.

Developers will see changes at all levels, though faster at the higher levels of abstraction. **While changes will be slower at lower levels, we will see better low-code developer tools emerge** (e.g. auto-complete, opportunities to refactor, more efficient code reuse,

AI/ML for code review and optimization). We'll also see **more low-code varietals – such as data science, chatbots, and AR/VR** – beyond websites, mobile apps and web apps. Low-code/no-code technology stacks will emerge for specific verticals or use cases. **It'll be a parallel universe to the world of custom development** – Makerpad, for instance, is aiming to be the "Github for no-code."

If we take a step back and consider the evolution of APIs and prebuilt integrations over the past decade, it's extraordinary what they've enabled so far largely in the hands of developers – just over 1% of the US population. **What it implies for what lies ahead – when we put these capabilities in the hands of dramatically more people – is exciting**. Big tech firms like Amazon and Alibaba rely on the connectivity of APIs to operate their organizations as platforms – to "datafy" every exchange, "software" every activity, keep data flowing, and apply algorithms to everything. **Imagine then the power of integrations bleeding into the rest of the world, into the nooks and crannies, until the digital world becomes the super platform**.

Low-code/no-code is also more than just APIs and integrations. **It can "codify" learnings across multiple dimensions** – for instance, through design blocks, business function workflows, and industry templates – to allow users to build upon other people's hard-won knowledge. **The low-code/no-code movement is also more than just doing the same things faster** – the pace of innovation is driving new approaches to how information artifacts can be used and useful. **The most compelling offerings are breaking existing boundaries and tradeoffs.** For instance, Notion, Coda and Airtable are blurring the lines among established document types and applications, while Webflow is both easy to use and full-featured.

We'll also see low-code helping to make AI and data science more accessible, rather than the realm of specialized researchers, big tech firms and large enterprises. Advanced technologies such as OpenAI's "shockingly good" GPT-3 language generator will find themselves in the hands of average citizens (perhaps as a drag-and-drop module in a low-code platform) – for better or worse.

This is the "gold rush" of the low-code/no-code movement. The removal of technical barriers to app development is bringing many more ideas to market for validation. People will be able to focus on where they can offer value, whether it's content or community or brand or something else altogether. **We will have a lot more software in the world and much of it will be lower-quality** – the democratization of video creation, for instance, gave us YouTube.

But some of it will be amazing. **In that flood will be solutions to a lot of problems** – real problems faced by real people, better served because the people solving them are closer to the problem. Some of the solutions will address big problems faced by a lot of people, such as pandemics and climate change. Some of the problems solved will be smaller, perhaps the niche needs of a fly-fishing group or local preschool.

It used to be that a problem needed to fit into a box of problems large enough for someone to invest in building a solution around it. Now someone who's not a developer can solve that problem for themselves and their own communities. **With more non-developers with diverse backgrounds and experiences involved, the possibilities frontier will be pushed out and the universe of what is possible expanded**.

Our thanks to Doc Williams (Founder/CEO, Brand Factory Inc; Build With Me) and Anthony Marcar (CTO, Kaddy) for their comments on this brief.

Disclosure: Contributors have investment interests in Microsoft. Amazon and Google are vendors of 6Pages.

Have a comment about this brief or a topic you'd like to see us cover? Send us a note at tips@6pages.com.

Team

Tam Thao Pham
CEO & Editor-in-Chief
tam@6Pages.com

Tam is an angel investor and advisor to early-stage startups. She spent over a decade at Deloitte, first in strategy consulting and M&A market diligence, then as inaugural Edge Fellow at the Center for the Edge, TMT practice co-lead in Hyderabad (India), CSO of the Global Office of Digital Innovation, and head of early-stage investments for US Consulting Innovation. She built and ran Global Innovation's Sensing function for 3 years, publishing the Line of Sight and briefing 3,000+ senior leaders on trends. Tam founded and ran a TEDx for 5 years and has been a regular TED-goer for 10 years. She has an MBA (UChicago), Masters in Organisational Psychology (London School of Economics), and Bachelors in Mathematics of Computation (UCLA). She lives in SF with husband and toddler, and is always on the hunt for new market shifts.

Dylan Hannes
Head of Content
dylan@6Pages.com

Dylan previously worked at Deloitte, first in Strategy & Operations Consulting and later in the US Consulting Innovation and Global Innovation groups. There, he advised executive leadership on new market shifts impacting firm innovation priorities as an author for the Line of Sight publication under Global Innovation's Sensing function, and also designed and managed a crowdsourced market-intelligence function. Dylan also helped author the best-selling book *Detonate: Why - And How - Corporations Must Blow Up Best Practices (and bring a beginner's mind) To Survive*. He lives in SF, and if he isn't writing, he's probably outdoors.

Eric Thompson
Head of Product
eric@6Pages.com

Eric spent 8 years in strategy consulting and innovation at Deloitte, where he helped create and scale its US Consulting Innovation arm focused on investing in and incubating digital products. He built and led a market intelligence service advising Fortune 500 clients on emerging trends and led a similar function serving 500+ Deloitte practitioners internally. He was Product Manager for Bridge, an online platform connecting startups with large enterprises for investments, partnerships, and M&A. Outside Deloitte, Eric started an email newsletter educating readers on global issues through digestible and unbiased narrative, and spent a summer as an investment banking analyst at Jefferies. When he's not working, he's likely at a concert somewhere in Austin.

Adam Tait
CTO
adam@6Pages.com

Previously, Adam was a senior software development manager and principal software engineer at Walmart Labs working on digital receipts and scalable infrastructure. Adam has also held prior engineering roles at Amazon, Qualcomm, Rally, and various startups. He holds a Bachelors in Software Engineering from the University of Waterloo. He lives in SF with his wife and toddler, and his favorite language is Clojure.

6 PAGES

About 6Pages

6Pages is a specialized market-intelligence service focused on far-reaching market shifts in business & technology.

The problem: Decision-making agents in organizations – the leaders and executives – lack time to gather enough context to make effective decisions.

The premise behind 6Pages is that the fastest way for humans to achieve understanding is through the narrative form.

We systematically scan thousands of market events, surface and connect signals of change, and run consulting-style sprints to build deeply researched briefs each week on what's happening and what it means.

These briefs, offered through subscription at **6Pages.com**, help professional decision-makers gain the context to make better and faster decisions.

"Leaders face the dilemma of either being a decision-making bottleneck or making the decision from the gut. Neither is a good outcome."

Follow @6pagesinc

Follow @6PagesInc