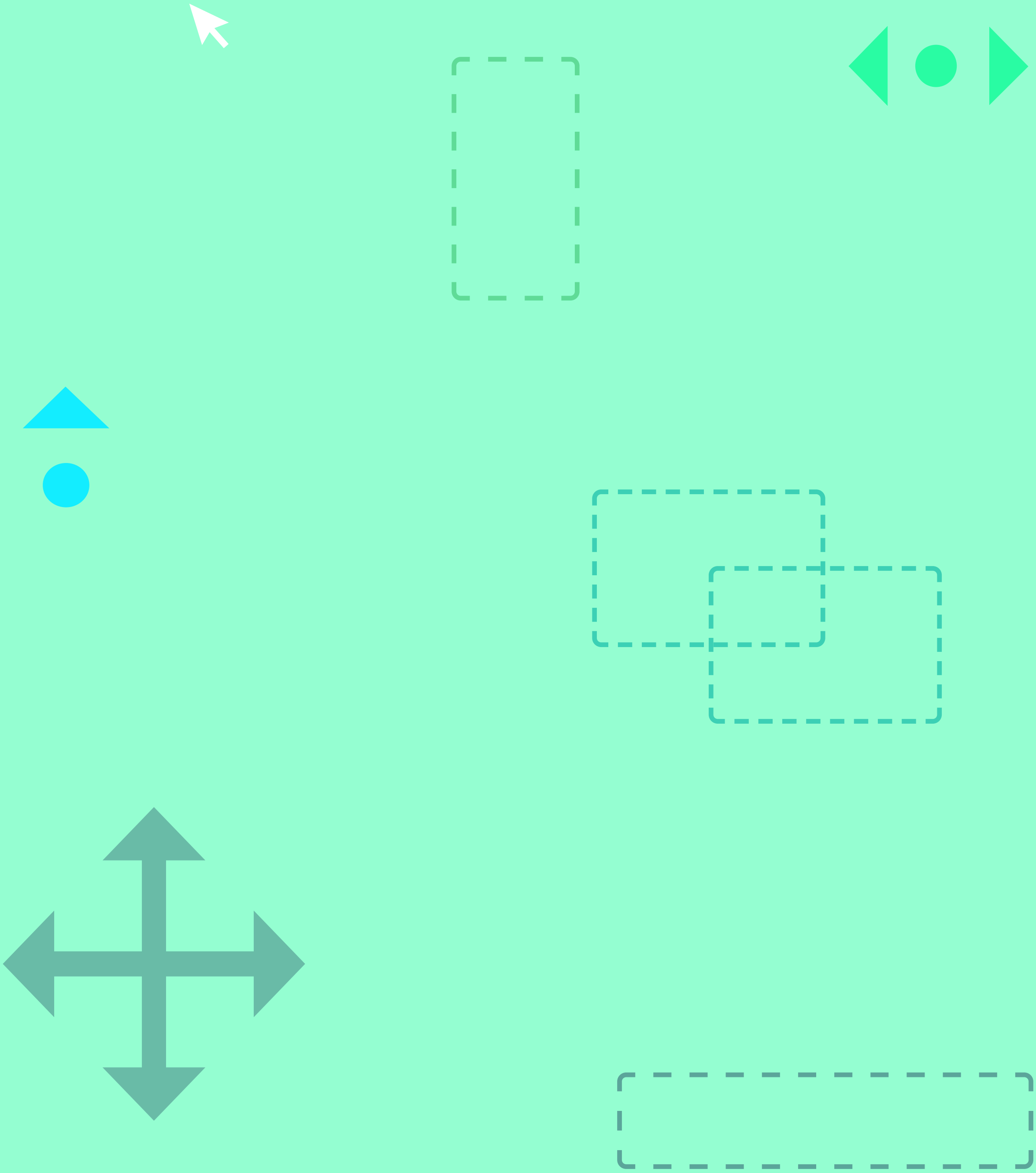
Neota

No-code Ebook

Empowering everyone to build digital solutions



Contents

ALL ABOUT NO-CODE

Section One: No-Code Overview

Section Two: Citizen Development

Section Three: Guide to Picking the Right No-Code Platform

WHAT CAN NEOTA OFFER

Section Four: The Neota Difference

No-Code Overview

What is 'no-code'?

Let's start simple. As far as naming conventions go, the term 'no-code' doesn't get more self-explanatory! No-code is a form of software development which allows users to develop applications without having to code it themselves.

There is code behind it, but it's now represented by visual and graphical drag-and-drop interfaces to give users the power to build digital solutions – without touching a single line of code.

This is the theme that runs through all no-code platforms; enabling everyone to develop software.

The history

It's easy to forget that software development is a relatively recent advance in the way we work. It's harder to forget just how seismic that impact has been. Every industry has been disrupted by software: in the words of Marc Andreessen, it has 'eaten the world'.

Software development, as with anything in life, is not static. The terms 'low-code' and 'no-code' are the latest stage in this evolution. While the words are new, the concept of no-code has been around for a long time. From the early days of spreadsheet programs like Lotus 1-2-3 and Microsoft Excel replacing the need to manually code software that can handle complex calculation to current webpage creators like WordPress and Wix where anyone can build their own business website, no-code platforms to build digital solutions are just the latest in this timeline.

Long before the term no-code arose, Microsoft Excel provided computer users with a way to replace manual manipulation, analysis, and visualization of data. Its benefits are still reaped today in practically every industry, long after its release in 1985.

In 2003, WordPress broke another barrier for businesses and organizations across the globe. Having that all-important online presence no longer meant having to pay a web developer to create a website. After a few hours with the system, anyone could have a functioning website for a low subscription fee. Even today, 35% of websites are powered by WordPress, with a significant proportion of the rest built on other platforms based on a similar model.

A multitude of other commonly-used no-code platforms exist today, and their popularity is rising fast. Tools such as Zapier, Google Sheets, Airtable, Notion, and Coda empower their users to build things they couldn't have done before without having to spend time learning to code. These platforms also provide far more features and possibilities than anyone could reasonably develop without having to spend a very large sum of money on development.

The crux of no-code is about the democratization of software development. In other words, it is all about accessibility. An interesting way to consider it, and why no-code is here to stay, is to think about it from a business standpoint. Possessing knowledge about coding has been found increasingly on job descriptions in the past few years, especially in professional services. But do companies really need someone who can code, or are they really after someone with the ability to develop software?

There is a subtle nuance between the two. No-code comes into the latter, offering a way of bringing software development down to a business user level. The democratization of the task means more people capable of doing it, without having to develop new competencies.

The phenomena of no-code platforms can be seen as a reaction to the rising need for coding; these new tools fix the problems surrounding coding and developing with agility and ease. The broader SaaS (Software-as-a-Service) industry is also underpinned by this idea – enabling users to do more by providing them with a software solution.

> **Even today, 35% of websites are powered by WordPress, with a significant proportion of the rest built on other platforms based on a similar model.**

No-code and low-code

So far, we've just mentioned no-code.

An important distinction to make is between no-code and low-code, two terms often found close together. You will likely see these terms more and more often, as Gartner predicts that the no-code/low-code market will be responsible for more than 65% of app development activity by 2024.

The fundamental difference between no-code and low-code tools is, again, accessibility. How easy is it for someone with no programming experience to use your tool? How much of your technology's features are accessible at the point of purchase to that non-technical user?

A low-code tool can offer businesses enormous opportunities for efficiency. It can eliminate the need for a lot of the repetitive, mundane coding necessary to build streamlined systems. But low-code isn't a panacea: these tools commonly need users to roll their sleeves up and do some more technical work to make the solution fit their business.

This assumed level of programming knowledge often also means that low-code technology providers tend to take a hands-off approach. You buy the technology and off you go – with minimal further support when things get complicated. Low-code platforms are often very powerful and can offer great benefits – but you must be prepared for a very steep learning curve - it isn't a tool that can be placed in front of business users. Low-code platforms often require the assistance of your IT team to make code changes for the end solution to run properly.

No-code, on the other hand, takes the entire concept a step further. It starts from similar principles: to make building digital tools far easier, far quicker and far more cost-effective. But no-code differs in some major ways. One is that no-code wants everybody to build digital solutions. It doesn't want steep learning curves, or baseline knowledge of JavaScript, or a mandatory 10-week coding boot camp.

The very real benefits which no-code/low-code platforms can provide to businesses are clear from this thinking. We've moved past the consideration that this form of development could be a bubble yet to burst, and instead realized its potential. Forrester Research expects the market for no-code and low-code platforms to grow from $3.8bn back in 2017 to a staggering $21.2bn in 2022.

> **Forrester Research expects the market for no-code and low-code platforms to grow from $3.8bn back in 2017 to a staggering $21.2bn in 2022.**

Citizen Development

What is 'citizen development'?

No-code platforms have given rise to a new breed of software developers, often referred to as citizen developers. The idea here is simple; an individual without a knowledge of coding can develop software, and this individual may also very well be the end-user of a particular solution. What is important to consider is how this isn't only a desirable new categorization from a business standpoint; it is also a necessary one.

The IDC predicted that the number of applications built by businesses between 2018 and 2023 would exceed 500 million; more than the previous 40 years combined. While on the one hand this could be seen as a result of a simple increase in demand, the fact that no-code and low-code systems now exist opens the door for many businesses to engage in digital transformation, many who could never have considered it before. Now with only a finite number of coders available, the reality is that people outside of this classical software development remit will often have to be the ones developing the apps. We can view this as a natural, proportionate shift in software development. The reality is that demand for digital creation is always outstripping the resourcing available.

This is no new phenomenon.

In 1982, James Martin noted this exact trend, but with the proportion of programmers to computers more broadly. In order to keep up with this trend in app development, this gap needs to be filled. But it is naive to consider the only solution to this as being to have a drastic rise in the number of people who can code. In parallel, for any real prospect of a sustainable future, software development needs to become more accessible. And that's what no-code and low-code platforms provide the ability to do.

The Enterprise Citizen Development Tension

Gartner defines a Citizen Developer as ‘a user who creates new business applications for consumption by others using development and runtime environments sanctioned by corporate IT.’

While not a bad definition, some have taken aim at the words ‘sanctioned by Corporate IT,’ reasoning that by its very nature, citizen development should encompass anyone building applications without the hurdle and oversight of the IT team. This tension should not be the adopted mindset.

A large part of this mindset lies in the simple fact that the ‘enterprise’ citizen developer hadn’t existed until very recently, and the practice of software development lay very resolute at the feet of the IT team. Any encroachment on this practice can very easily – at the beginning at least – be seen as a threat.

The fundamental issue rests in the ambiguity surrounding enterprise citizen development. How far does their reach go? Do they build the apps and make them live for the end-users themselves? Do they maintain the apps and patch bugs? All these questions pertain to where programmers and IT departments would sit in a world where individuals who don’t code are building apps.

The suggested mindset

Although platforms that drive citizen development should be business-led purchases, IT oversight on the purchase and ongoing engagement in its use is critical to ensuring success and avoiding the resulting trap that is shadow IT.

Business-led demand for no-code/low-code platforms should not arise from seeing IT as a speed bump in getting things done. It should be seen as a need to encourage downstream, non-IT development on work that doesn't need resourcing from what is often an understaffed team. Procurement and subsequent use of a no-code/low-code platform should always be sanctioned by IT – the real value lies in the resulting freedom to ideate, develop and deploy applications by business users without needing to add to IT's pre-existing burden. After all, IT cannot do everything.

Let's consider this with the example of a law firm wanting to develop a client-facing app without the use of a no-code tool.

There are almost always inconsistencies between what the Subject Matter Expert (SME) wants and what is being built by the development team. Following one iteration, the SME might change their mind, or want something added in, or a UX change, or any other of a variety of potential issues. Note that this isn't to blame either party; if the SME doesn't know how to code, they cannot explain exactly what it is they want, how to bring it about, and how it should operate.
On the other side, there might be delays in the SME explaining what they want, and there is an underlying assumption that the developers would understand the complexities of particular areas of law and understand the terminology used and how it applies in building an application. Time and money are wasted on fruitless reiteration processes.

Now consider if the SME was able to build the solution themselves, and develop it exactly how they wanted. There is still a possibility of issues arising or the SME not understanding fully how to do a particular thing, but the IT department is liberated from the vast majority of the excruciating back-and-forth process.

> **Procurement and subsequent use of a no-code/low-code platform should always be sanctioned by IT – the real value lies in the resulting freedom to ideate, develop and deploy applications by business users without needing to add to IT's pre-existing burden.**

Let's consider this in a broader digital transformation sense. Many technologies aim to free professionals from tasks that can be automated and simplified, with the ultimate goal of allowing those professionals to focus on higher valued work as opposed to, for example, redrafting the same document 100 times.

No-code does the same thing for the IT department. Certain development work can be done by the business, more easily, with less reiteration, saving time and resources and focusing the expertise of the IT department where it is needed. It empowers the IT department to focus on projects that require absolute attention, with fewer resources needlessly dedicated.

Citizen development as an enterprise-level solution

What this discussion is building to is a view of citizen development being an enterprise-level solution, assisting a variety of business functions effectively.

Citizen development is about bringing about an enterprise-level capability to drive digital transformation, and develop a variety of solutions as and when they are required. It allows us to set aside the classically centralized idea of software development and of who is capable of building. Rather than either buying into a prescriptive point solution or having a bespoke app built by your IT department or external developers, the possibilities of app development are made far more accessible and provide the opportunity to have more than just one solution for a single price.

This isn't just an abstract forecasting, either. If you consider that nearly 60% of custom apps in North America are now built outside of IT departments, by individuals with little to no coding experience, the magnitude of digital transformation and citizen development becomes clearer. And it is important to note that this doesn't have to exist in isolation. Citizen development has to form at least some part of a digital transformation or enhancement strategy for businesses to sustainably develop their service delivery.

Guide to Picking the Right No-Code Platform

Platform solution vs. Point solution

The very first question you need to answer is: are you looking for a point solution or a platform solution?

If you've answered the latter, then ask yourself: are you looking for a platform solution to solve a specific problem or a strategic pick to complement your organization's broader technology stack?

Let's dig a bit deeper.

Point solutions enable you to tackle a single, very specific problem in isolation, allowing for deep functionality of one channel. One key disadvantage is that if all problems are solved with a point solution, it becomes very difficult to manage the multiple systems and databases that support them. Systems are created and launched in isolation, which can often lead to user frustration as each system brings a new interface and learning curve. It is also usually more difficult to expand and enhance the solution to solve other problems, making it less adaptable to changing business requirements.

Platform solutions enable you to address a number of use cases, across multiple facilities with integrated and consistent end-to-end business processes throughout the platform. They are a logical choice if you wish to expand your application, existing technology stack or address several use cases in the future. However, they can often involve reimagining solutions to help fit within the platform's capabilities – where there are major gaps though, integrations with other platforms can often help solve these challenges.

Often, change management is centered on solving the immediate problem, and so often look to point solution amendments, focused on keyhole changes. No attention is paid to downstream (or upstream) effects from these changes. Utilizing a no-code approach to transformation requires an end-to-end look at suitability and fit.

What to look out for when choosing a no-code platform solution

There are a number of considerations to be made when looking at a platform's features and capabilities, and it can be a minefield trawling through the phrases and acronyms used by vendors when establishing what is of value to you, your team and your business. A good vendor will help you untangle this and work with you to derive value for your business, from the platform.

No-code platforms exist on a spectrum. On one extreme you have platforms offering very basic functionalities – i.e. simple form and logic creation, combined with rudimentary document automation capabilities. On the other, you have platforms allowing citizen developers to build large, end to end workflow solutions, encompassing features like e-signature integrations, multi-step approvals, email reminders and data management.

The considerations we will run through below seek to help give you a clearer picture of what you are investing in, and how to approach the buying process when it comes to no-code platforms.

Pre-built use case templates

No one really wants to start with a blank canvas when building anything. Solutions built on no-code platforms are no different. Users look to no-code platforms to cut down development barriers – both time and resourcing.

The frequent homogeneity of particular parts of applications means that some vendors now offer pre-built use case templates, offering either a complete simple solution or a core part of a larger, more complex solution. In any case, these building blocks help you by allowing you to simply drag and drop larger parts of an application in a modular manner as opposed to developing it yourself. While developing it yourself in a no-code platform shouldn't be a hard task, these templates empower you to speed up the app-building process, allowing the user to develop comprehensive end-to-end solutions with less investment in time and energy.

While many vendors now offer this functionality, a question you should ask should be how comprehensive these pre-built use case templates are, and how far along the app building process they will get you for the use cases you have in your pipeline.

Process automation

Workflow modules are becoming cornerstones of most no-code platforms. The ability to allow no-code development of solutions that allow for tasks that span multiple parties or time periods, or where complex negotiation or signature sequences are required opens up a world of possibilities for users.

The scoping of workflow solutions will always be a visual exercise – think Visio for example. Therefore it is critical that the no-code platform you're evaluating allows for visual drag-and-drop creation of your workflow solution or allows for import from another tool. The additional advantage of being able to scope it out in your chosen tool is that you can better understand the constraints of the system when mapping out your solution. This allows you to piece together an application with a greater understanding of what is going on, and what you want the system to be doing. Having the tools aligned to a widely recognized standard (i.e. Business Process Management Notation (BPMN)) is a major advantage.

Beyond this, the platform should be able to demonstrate how the automation could speed up, enhance, and simplify day-to-day tasks, and the complexity of the processes that they can undertake. Useful tools for enabling this type of automation include having sequential, parallel, and loop task creation capabilities. These are especially useful in process-heavy use cases, where service requests and approvals may need to be made. The ability to automatically loop back in the process if approval is denied, or sending relevant information off to another department or individual in a particular circumstance allows often messy and admin-heavy tasks to be fully automated with ease.

Other useful workflow-oriented capabilities include 'optional' action creation, allowing individuals to edit, update, and assign aspects of a task, as well as role-based credential access and management control to databases.

Most importantly, understand how role-based credential access and management works on access to the end solution, not just the platform.

One of the most useful offerings can, however, be data visualization support. This is particularly relevant where an application is dealing with one or several databases from which the system is extracting data. The ability to automatically see your data in a clear, digestible format again allows a business to speed up processes and understand their processes in greater detail without having to manually create data visualizations.

Document automation

While many vendors offer document automation capabilities, the ways in which they function and their interface can vary significantly. This means that rather than just ticking document automation off the list of things a particular no-code solution can do, you should investigate how it works, and establish whether it is the right tool for you.

Questions you should be asking include whether the system offers document mark-ups natively in Microsoft Word, or in a separate interface. If it is unable to do this natively in Word, consider if the functionalities that your documents need in Word and ensure they are supported (e.g. formatting and clause referencing). For non-Word programs, also consider if the editing program
has a familiar and user-friendly interface, if there are any compatibility issues, and the overhead of using two different programs.

Even if the program is surfaced in Word, it's important to confirm support for Word functionalities in the automated document. For example, will automated numbering, cross-references, and footnotes update if clauses are added or removed, and whether you can leverage custom styles. When dealing with data spreads that require document automation, the ability to create and populate tables and other data visualizations automatically can be very important. It is also important to establish whether these tables and visualizations can be auto-populated with data in the form which you store that type of data in, pertaining to file and database types.

For document and contract negotiation, again there are a number of implementation possibilities. A system that has a comprehensive multi-user tracking, commenting, and editing system integrated into the document automation is often the core functionality but it's also worth considering more fixed negotiation processes such as fall-back clauses. Further, it is worth considering whether multi-party execution clauses are supported, and support for your chosen e-signature provider.

Expertise automation

An expertise automation capability is something that very much pertains to the spectrum of no-code platform offerings mentioned earlier. There are various levels of complexity offered by different vendors, and the important thing to look out for here isn't just high-level expertise automation capability, but a broad range of these capabilities, from simple to complex. The reason for this is that the majority of apps will use some expertise automation functionality, but are unlikely to require AI or other complex logic systems integrated into them. Key things to look out for include whether the platform offers simple, stand-alone logic creation, or support for combining different forms of logic in the same application. This can include 'if-then' rules, decision trees, formulas, and weighted scores. Further, you may want to establish whether you can combine declarative and procedural forms of reasoning in the same application for the purposes of building more complex logic-based applications. Another level up, some offerings may include reasoning across field sets, or even nested field sets, to support logic application across groups of data points for really complex applications.

It is useful to take a bit of time when establishing which of these are relevant to you, and what aspects of expertise automation functionality you would need in your identified use cases as well as potential future use cases. The offerings and applicability can be quite difficult to comprehend at face value, so getting a grasp of previous app examples built in a particular platform can help analogize your use cases.

Database integration

This is quite possibly the most important consideration for you to make when picking a platform. Ultimately, any digital transformation process seeks to utilise both user input and workflow data in some way.

Some platforms enable storage of application session data to external SQL databases, and most importantly, accessibility to you, as a user of the SQL tools, to these databases. However, this often requires database administrators or knowledge of SQL language.

Is it really a true no-code platform if a user still has to interact with an internal database administrator when looking to manipulate data or interact with SQL databases? Can you truly call a platform 'no-code' if you are still required to have knowledge of SQL operations?

Ask these questions – does the platform:

- Enable storage of application data in a SQL or Postgress database through no-code?
- Enable no-code, no-SQL interactions with standard, external SQL database?
- Enable automatic creation and update of databases when the application data or structure is updated?
- Enable automatic synchronization of application data structures?

Simplicity in scaling – the non-technical considerations

All SaaS offerings come with some degree of 'scalability'. This comes in many forms, but most frequently in how the subscription pricing ultimately scales. There are a variety of aspects to this to think through. Some of these relate to the internal building process, others more broadly on your no-code strategy. This may, for example, include considerations on whether to commercialize the app yourself. More broadly, ask yourself the question: does the platform's scalability align with our strategy going forward – after all, you shouldn't be buying a no-code platform to build one single solution.

Number of users

One aspect to pay attention to is whether the platform makes a distinction between builder users (i.e. those who are using the platform to build) and solution users (i.e. those who are using the end solution) and does the pricing scale based on an increase in this distinction. How many users and types of users you can have when building solutions and when the application is running. How many builders can you have working on the application? Is there a limit to the monthly usage of your application at runtime? Will you have to pay an extra fee or get a higher subscription band in order to have the app used at the scale you want?

Number of solutions

Beyond this, you want to see how far you can make your subscription cost go with the development of multiple applications for a variety of use cases. Does the platform scale based on the number of solutions being built or being used? If so, what are the costs associated with this? Does creating more complex applications lead to higher costs for that application? Consider also your database usage, as you will need to establish what data storage capacity a particular offering provides, and what the cost of expanding this would be. These are all fundamental questions when establishing your ROI in depth.

Commercialization

Question whether the vendor charges extra if you decide to commercialize the solutions you have built (i.e. if you are a law firm and you've built a solution for your client). If yes, how do they charge (i.e. revenue split or extra per solution that is commercialized)?

Support and training

Support and training are often left as the last consideration on the list – always after the commercial points. However, when it comes to no-code platforms, it should be somewhere at the top of the list!

It is crucial to understand what level of support and maintenance is included in your subscription model. Enquire about the types of support included in the base subscription rate, what extras are available, and whether there is a dedicated customer success and project management team to help you on your no-code journey.

An important consideration is understanding how the vendor charges its professional service hours for support purposes. While relying on the vendor to debug a solution or guiding an internal team on best building practices is often faster and more effective than 'figuring things out yourself', this can often be accompanied by unexpected line items in the monthly invoice. Ensure these charges are understood prior to contract signing.

Along with support, understanding what user training is included as part of the onboarding process is essential. No matter what no-code solutions you are looking at, there will be some level of training required to get you to where you need to be.

Considering many organizations haven't been on an internal app building process before – or it is taken care of by IT rather than your citizen developers – a certain level of hand-holding is always helpful coming from the vendor to guide you through best practices and how to approach your digital transformation on a broader level.

This helps to provide a deeper level of understanding of the process, and ultimately provides for faster and more confident app-building processes in the future. Some vendors may offer bespoke ideation sessions and design-thinking sessions to help you map out the framework of your application. This is invaluable knowledge when seeking to transform an idea into a functioning application.

Part of this support and training consideration is determining who will actually be trained in your organization. The user profile of the individual(s) managing and building solutions is key to how successful and efficient any app development process is. It is likely that particular individuals in your team are more suited to app development, not only in terms of their technical capability and the time they have to be upskilled, but also how well they can articulate specific business processes and logic flows.

An example could be where a law firm is seeking to enhance their internal correspondence and matter management. A lot of this is work is down to paralegals as opposed to fee-earners. While fee-earners may be enthusiastic to have their input – and it is important they do have some input into the planning – it may make more sense for the app development process to have the paralegals front and center in the process. In larger organizations, consider forming a Centre of Excellence (CoE), a centralized team who can leverage the platform for the rest of the organization.

The Neota Difference

Now that you have a better understanding of no-code and what to look out for, let's see what makes the Neota platform stand out from the rest!

The complete package

The Neota platform is best described as:

> **The only no-code platform that incorporates BPMN-aligned intelligent process automation that is further supported by expertise and document automation and data management capabilities.**

The platform provides professionals with pre-built Use cases to rapidly build digital workflow solutions that automate any aspect of their service, without writing a single line of code.

Now let's break that down a bit!

Process automation

Workflow is Neota's drag-and-drop, standards-based feature that allows users to create process automation solutions. Workflow enables organizations to manage tasks and complex processes that span multiple teams, parties and time.

Utilizing Business Process Modeling Notation (BPMN) standard visual conventions into a drag and drop interface, users of Workflow are able to orchestrate individual or multiple Neota applications, as well as external systems, in a logic-driven, data-aware end-to-end solution to manage the execution and completion of business processes such as:

- Document creation that requires conditional approval routing;
- Document execution that requires multi-stage negotiation and review;
- Case management and matter intake that requires assignment, tracking and review;
- Central tasks allocation such as employee onboarding or IT Support requests.

Further, this feature allows for the no-code creation of specific workflow functionalities such as:

- Conditional logic (do task A if X, otherwise do task B or send the next incomplete task to person X when Y is true)
- Reminders (if Task A has not been started or completed on time, send tailored reminders to the assigned people)
- Multi-party interaction (move work such as review and approval back and forth across parties and business unit
- Actions (resume a process, cancel a step, query an external database, reassign a task)
- Data management and audit capture (progress, status and all data elements are automatically updated as processes run. All user input and workflow data are automatically captured and presented in our case manager – a grid view of process sessions that users and managers can use to track progress, move processes along and gather statistics on outcomes, efficiency and cycle times. Visualization across this data is also possible through the Neota platform.

FEATURE HIGHLIGHT – WHITEBOARD MODE

Going one step further, the Workflow designer has a 'Whiteboard Mode' which allows a subject matter expert to define how the intended process should flow before any application building has started. They can do this with no Neota training.

What this means from a practical perspective is that you can ask stakeholder(s) familiar with the business goals and processes to define the workflow in the Whiteboard and when completed, can pass this onto a Neota trained business user to complete the actual build. With content and knowledge being the biggest barrier to building any platform solution, this takes the tools right to the content person!

Document automation

Neota's Word Toolbar feature increases efficiencies and control over automating templates created in Microsoft Word. Through the Word Toolbar, Neota authors are able to create questions, conditional text and logic variables, while supporting all existing features of the document (tables, headings, footers, bullet points etc). This also includes the automated handling of clause references which is an often overlooked functionality by document automation vendors.

These questions, conditions and variables can then be automatically imported into Neota's authoring environment, to create the framework for an end-user solution that controls the document creation as part of a larger Contract Lifecycle Management workflow solution.

Negotiation capabilities are further supported by the Neota platform through native redlining, departure tables and clause libraries that enable the negotiation of document content across multiple parties.

Expertise automation

Neota's expertise automation function is driven by our proprietary AI inference engine. The engine employs hybrid reasoning tools capable of emulating the 'it depends' reasoning often found in legal analysis. In building an application, all content/rules/factors/judgement that a subject matter expert would consider in making a particular judgment call will be automated so that the app will provide the same advice the expert would have made, given the same set of facts. Unlike 'black box' systems, we can provide a report on how the decision was reached, thereby creating a complete audit trail for future reference.

Our inference engine allows the creation of highly complex applications - for example, a regulatory compliance checking application that instantly processes millions of fact patterns to assess compliance applicability or commercial risk against thousands of market rules.

> **The engine employs hybrid reasoning tools capable of emulating the 'it depends' reasoning often found in legal analysis.**

Along with the inference engine, the platform also encompasses a range of other tools that allow users to create complex logic-based solutions. These tools include:

- An advanced Formula Editor which supports the construction of highly sophisticated calculations and reusable functions to support automated financial outputs;
- A Spreadsheet Editor that allows existing Microsoft Excel spreadsheets to be uploaded to a Neota built solution where that solution requires calculations that are best (or can only be) completed in Excel. The editor allows platform users to maintain sophisticated calculations or models natively within Excel spreadsheets, with the operations to serve as an additional reasoning tool for the builder. Intuitive web-based application interfaces can then mask the complexity behind the scenes and save users from editing Excel spreadsheets directly.

These schemes allow you to improve consistency in decision making, streamline expertise delivery, and scale your expertise in a way that is impossible to recreate manually.

FEATURE HIGHLIGHT – CANVAS

Of course, not everyone will want to develop a solution with such advanced functionalities. Neota's expertise automation capabilities are further enhanced by our prototyping tool - Canvas. Canvas is a web-based authoring tool that enables subject matter experts to quickly and effectively turn their expertise into runnable applications. Canvas incorporates the legal engineering or process mapping steps an expert would typically undertake when mapping out the logic of an application. Similar to our Workflow designer, it replaces third party process mapping tools, to significantly streamline the way clients can turn ideas into runnable products.

Canvas requires no training and is a logical, intuitive product that allows users to ask questions, apply formulas and rules, and generate output reports or send emails to third parties. All Canvas outputs are fully 'ingestible' into Neota's main authoring platform – Neota Studio for advanced functionalities (think database connections and integrations, advanced logic and reasoning) to be built out.

Our vision is that Canvas will be deployed entirely across an organization, to give every employee, whether you are a lawyer, HR, procurement professional, etc., the chance to start prototyping ideas for automation. These prototypes are then put through an ideation funnel to select the best few, which is then handed to a separate team of trained and certified Neota builders to build out the solution in the Neota Studio.

Data management

The Neota Data Manager (NDM) is a radical innovation in data management.

With NDM, business users are empowered to build applications that read from and write to MSSQL and Postgres databases, without writing code and without assistance from programmers or database administrators.

NDM can automatically generate a database with all the tables, joins, and procedures needed to manage data for a Neota application. As the application's data structure is changed (variables added, modified, deleted), the database schema is automatically updated and synchronized, meaning that changes to any Neota built solutions requires relatively minor overhead.

Operations to read from and write to the database are created with a few clicks via the Neota authoring platform. Most importantly, knowledge of SQL is not required.

All actions taken in applications on the Neota platform, including user or logic-driven amendments to contracts, are automatically stored in NDM and are readily available for reporting purposes or to trigger actions based on pre-defined logic or other criteria. Configuring NDM is flexible, and an organization can either use Neota's database or one of their own.

If using Neota's own MSSQL or Postgres database, all data is stored in a virtual private cloud, maintained for the customer by Neota at Amazon Web Services, Azure, or in the customer's own cloud.

Where the customer has an existing database – for example, a customer relationship management system – NDM can read the database schema and build the Neota application data structure automatically, and without code.

Use cases and building blocks

No-code platforms, like with most platform solutions have the blank canvas dilemma. The catch-22 that while everyone wants the tools to be creators, no one wants to start from ground zero – no matter how good or easy the actual technology is to use.

Even the Microsoft suite doesn't escape from this problem, and they've implemented templates to help solve this. For Microsoft, a PowerPoint Template is 'a pattern or blueprint of a slide…[templates] can contain layouts, theme colors, fonts, effects styles and even content.' The building industry has something similar – the Modular Construction method. A form of prefabricated construction methodology that sources pre-built components (made offsite) which are then transported to the construction site to be assembled. In essence, it's a lean technique applied to the construction industry!

To solve this problem and to enhance the platform usability, we have introduced our use cases and Modular Building Block concepts to the Neota tool suite!

There are three key concepts:

Use cases are fully built, runnable digital solutions which are made available to Neota customers. Although pre-configured by the Neota team, each minute use case detail (logic, functionality, content) can be configured by customers to suit their own needs.

We have included two use case data sheets as appendices to this e-book. For a complete list of use cases, please visit our website.

Use case templates are skeleton versions of use cases that Neota customers can access to customize or further develop through the Neota platform. Use case Templates do not have content built-in, rather it comes with step-by-step instructions for the customer to leverage already built out functions of the use case while allowing customization of their own content.

Building blocks are modular, fully built, meaningful, 'functionality specific' solutions which through simple no-code 'plug and play' can be connected to use cases or use case Templates. Building blocks include functions like third-party integrations into DocuSign and Box, negotiation capabilities and approval workflows.

Integration –
What does the system support and allow you to integrate?

The ability for any newly onboarded tool or platform to integrate with existing systems is critical to the enterprise success of any technology stack.
The Neota platform provides a native Web Services editor that functions as a visual, no-code component to make REST API calls to third-party products over HTTP and secure HTTP (HTTPS). As such, it allows integrations with any product exposed via a RESTful API.

This empowers builders to create their own integrations, so there is no need to come to Neota to custom develop integrations for each platform individually. To help give a better understanding of what this can allow you to do, successful third-party integrations with Neota include DocuSign, SharePoint, Salesforce, HighQ, Kira, Wolfram, AWS, Google Maps, PowerBI, BigML, iManage, GreenID, TextRazor, Dandelion, and Amazon QuickSight, amongst many others.

The ways in which these integrate allow reasoning tools to have their data points translated into the Neota platform, allowing for reports created at the end point to use that data and apply any necessary reasoning to it. Other functionalities, such as DocuSign, allow for Contract Lifecycle Management solutions built using Neota to be integrated with signing procedures in one native user experience, rather than having to link in and out of the Neota application in order to achieve these ends.

Neota's integration capabilities extend in the opposite direction as well. Neota applications can be run as a service to other Neota apps or to any program or external third-party system via the Neota-as-a-Service (NaaS) mechanism. NaaS provides a mechanism for Neota built applications to be transformed to expose a RESTful API for integration to third-party systems. For more information, please read our Integration Whitepaper found here.

Security

Data security and privacy considerations are never an afterthought here at Neota. All data used and maintained by the Neota System (NLS) is encrypted in transit (via HTTPS/SSL) and at rest (through continuous AES-256 encryption of the NLS Operational Database). Neota Data Manager (NDM) data stores can be located on customer premises or in a customer-controlled cloud (at AWS, Azure, or another cloud provider). Customers hosting an external data store will have complete control over the encryption of that data.

Neota applications can persist data via NDM to Microsoft SQL or PostgreSQL databases, or via the NLS Web Services Editor to any data store or system with a REST API. NLS application servers and relational databases are entirely contained in an AWS VPC, making it impossible to directly access these machines from the general internet.

NLS environments are protected by Amazon Guard Duty, an intelligent threat detection service that continuously monitors the network for malicious or unauthorized behavior. AWS Shield protects against the most common Distributed Denial of Service (DDoS) attacks and state of the art web application firewalls help protect the infrastructure from common web exploits that may affect availability, compromise security, or consume excessive resources.

Manual and automated threat detection is performed by Neota developers and third-party services several times a year, with feature updates (and security patches) being constantly tested and debugged prior to their release on customer servers.

Solution development

Neota offers our customers the choice when it comes to the development of applications in the platform – self-service or Neota managed service delivery. Both options are supported by our customer success team during three key phases depending on the use case that is being developed and who the end-user will be, they are:

1. Idea to prototype – The process of managing business ideas/challenges/pain points through a 'prototyping funnel'
2. Prototype to product – The process of project managing a set of deliverables into a working product and releasing to the business
3. Product to market – The process of taking a product to market

As part of Neota's onboarding process and to support customers on their Neota journey we can offer to run customer workshops that support design thinking, ideation and agile application building.

For customers that want to use Neota Service Delivery we take a hybrid approach to project managing application delivery which incorporates elements of agile working.

Self-service/Build your own applications

Customers that are looking to be self-sufficient in using the platform and can build and maintain their own applications may take one or more of the following approaches:

- Self-paced e-learning
  Neota provides both the Neota University (more information below) and User Manual for customers to use training and reference material. The self-paced e-learning material enables customers to become familiar with the fundamentals of Neota Studio.

- Training workshops
  – 2-day Foundation or 3-day Standard workshops for attendees that will be using Neota, which are designed to bolster participants' knowledge of Neota, as well as work with the participants to develop a working Proof of Concept application based on an initial use case identified by the customer.
  – 1/2 day Advanced workshops which are designed to focus on very specific advanced platform capabilities, which may only be required for certain use cases.

- Bespoke training workshop
  If a customer has very specific training that they would like Neota to deliver, we would be happy to discuss and under the customers' requirements and expectations. If the training material does not exist and/or the delivery of training requires a Neota specialist to deliver, this will incur additional cost to the customer in preparing and delivering the required content, along with any associated travel & expenses.

Neota managed service delivery

Neota Service Delivery is based upon standard project management methodology, a framework to support the successful delivery of an application/product or solution to our customers who have elected not to build using the Neota platform.

Neota projects are delivered in 6 phases (planning, solution blueprint, build & test, customer user acceptance testing (UAT), deployment and support transition), with each phase having a set of tasks and outcomes to complete.

During the build & test phase, Neota uses an agile approach; building being carried out in 1 or 2-week iterations, with a customer demonstration at the end of each iteration. This allows continued customer engagement and proactive feedback during the build process. Feedback is logged in a product backlog for review and inclusion in a future release. Certain phases have a tollgate to move through before proceeding to the next phase of delivery, which acts as a formal authorization/signoff by the customer to proceed to the next phase.

Neota University

Neota University is Neota's comprehensive e-learning and training platform. Neota University combines step-by-step training videos, clear reference materials and hands-on, real-world application building exercises to familiarize authors with the many functionalities of the Neota platform. Neota University provides detailed instruction on some of Neota's most exciting functionalities including Workflow, document automation, data storage and much more. Neota currently offers two curated courses for authors: Fundamentals and Advanced Concepts.

Neota Fundamentals

Fundamentals is designed for new authors looking to develop their app-building skills with our advanced app-building application-authoring component tool, Neota Studio. The course covers the basics of application authoring in Studio and guides learners through the process of building out their very first app. By the end of the course, authors will have the opportunity to submit their own Neota Expert Advisor application and receive personalized feedback and guidance from the Neota team.

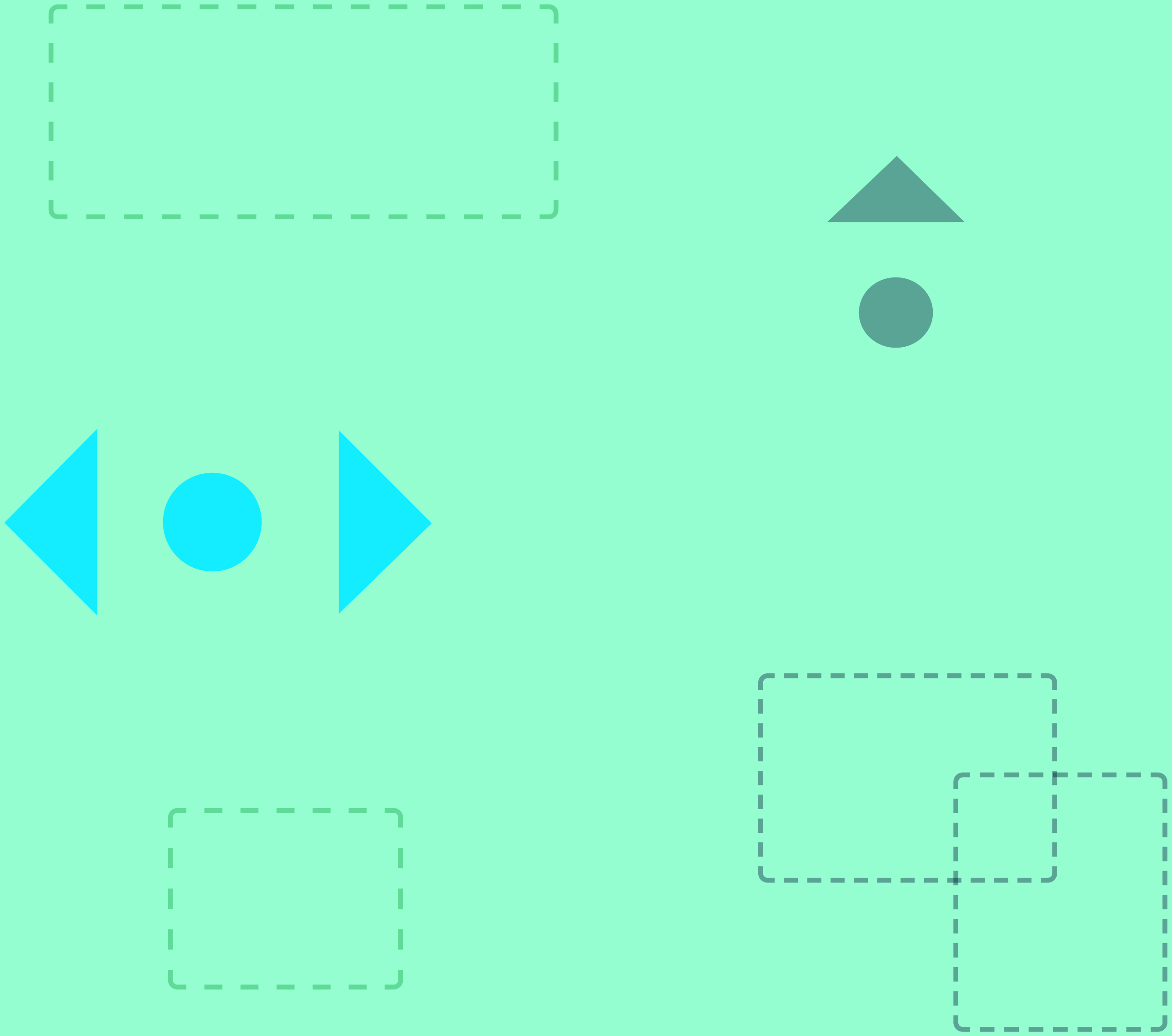
Neota Advanced Concepts

Once authors have completed Fundamentals, they can transition to Advanced Concepts to gain a wider understanding of Neota's advanced functionalities including Workflow and NDM. This course will teach authors how to leverage Neota applications to generate documents, how to create complex multi-user and multi-application Workflows, how to integrate SQL data storage into their business solution, and much more. Advanced Concepts will give authors the skills and confidence to implement sophisticated business solutions leveraging the full power of the Neota platform.

Certifications

Both Fundamentals and Advanced Concepts reward authors who complete the course with a certified digital badge and certificate of completion to recognize their achievement. Badges and certificates are digitally shareable and can be added as a verified certification on LinkedIn or shared via social media.

Thank-you for taking the time to read this e-book and we hope you are as excited about the future of no-code as we are at Neota! If you would like more information or wish to discuss any aspect of this e-book further, please contact info@neotalogic.com.



ABOUT NEOTA

Neota is the leading no-code platform for the intelligent automation of processes, documents, and expertise.

The platform provides professionals with pre-built components to rapidly build intelligent workflow solutions that automate any aspect of their service, without writing a single line of code.

These solutions empower professionals to deliver access to their expertise at scale.

Learn more | Request a demo

Neota