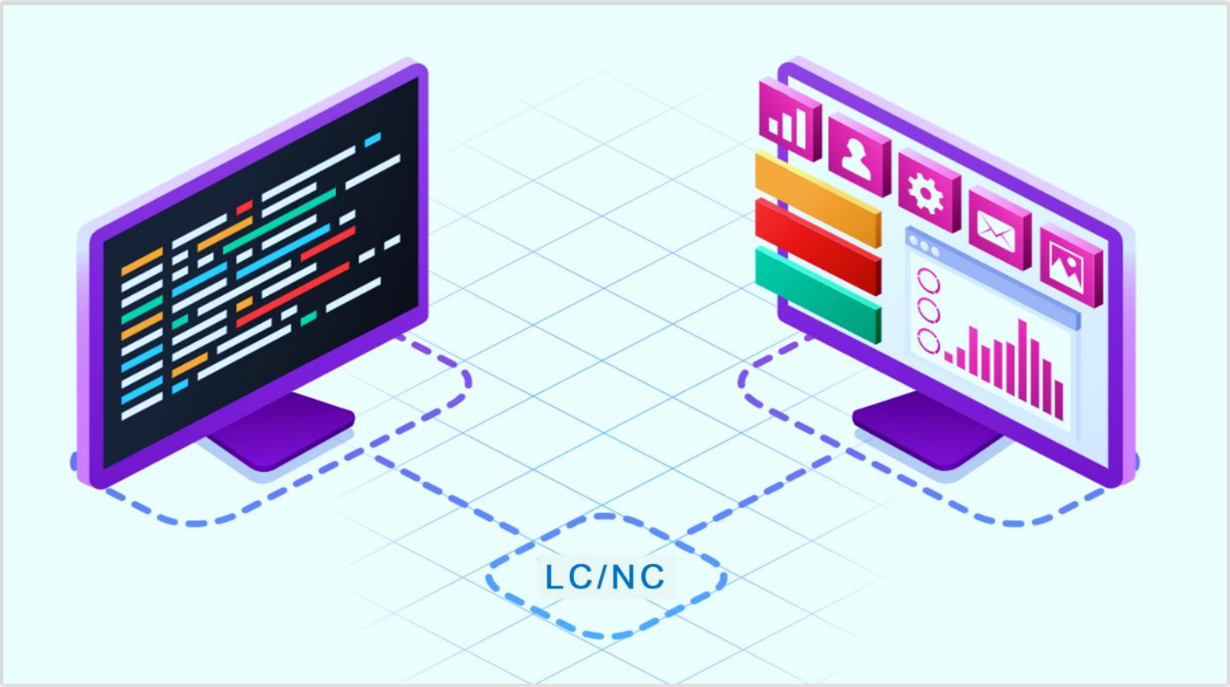
Comparative study of three Low-Code No-Code Frameworks

From the perspective of an SAP environment



Business Internship

Master:	Science & Business Management	Intern:	Roland Weenink
University:	Utrecht University	Daily supervisor 1:	Ananya Chatterjee
Company:	Accenture	Daily supervisor 2:	Aman Garg
Duration:	01/05/2023 → 16/11/2023	Reviewer:	Soumalya Nath
Date:	07/11/2023	Second reviewer:	Bart Verkade

Abstract

Constant pressure on businesses to innovate and improve efficiency in order to remain competitive has led to a shortage of software developers. This in turn has led to an increase in popularity of Low-Code No-Code frameworks (LCNC). LCNC frameworks are easy to use, visual drag-and-drop development platforms that enable employees with domain knowledge to develop applications without a proper programming background (citizen developers). This study assessed how three different LCNC frameworks, SAP Build, Mendix and Neptune compare based on a quantification method of their feature set. Furthermore, the strengths and weaknesses of these LCNC frameworks are determined and analysed based on their impact on the IT world. The findings show that Mendix has the average highest potential based on its features, having the highest collaborative development support and the most graphical user interface features. SAP Build has the strongest interoperability support, and the strength of Neptune is within the reusability feature. The strengths and weaknesses of each LCNC framework differ, therefore the impact of each LCNC is dependent on the specific use case. A recommendation to Accenture is made on how to position LCNC frameworks to their clients based on this research. LCNC framework adoption leads to the overall benefits of an increased development speed and cost reduction and drawbacks in customisability in general. The recommendation for a specific LCNC framework highly depends on the needs and sector of the client.

Management summary

LCNC frameworks can potentially drive innovation and reduce dependency on professional software developers. The value of LCNC frameworks comes mainly from the fact that employees with domain knowledge no longer need to relay their needs to developers. They themselves can develop an application or process fitted to their needs. However, the true value of LCNC frameworks and by extension citizen developers is highly dependent on the specific challenges, priorities, budget and needs. The benefits of a correctly implemented LCNC framework within the business strategy can increase the development speed of applications up to 56%, reduce the general costs of the applications by 53% and by extension increase the revenue of the application in question.

To be able to state which LCNC framework is the best to use is not possible, since different LCNC frameworks excel at different things. The quantification method of this study does indicate that Mendix is the most well-rounded LCNC framework in this study. If the three different LCNC frameworks need to be boiled down to their direct strengths, Neptune is best for quick, simple and sleek app design. For overall reliability, functionality and collaboration Mendix is better. For using a LCNC framework within an already established SAP System, SAP Build is best.

The position Accenture should take regarding LCNC frameworks to their client is a supportive position. Accenture should acquire in-house knowledge on development with LCNC frameworks, and support clients with educating citizen developers. Furthermore, Accenture can assist clients with setting up the LCNC frameworks within their business and take away the reluctance to adopt. If managed correctly, Accenture clients could gain a competitive advantage by adopting LCNC frameworks now rather than later.

Table of Contents

Contents

Introduction

Current business landscape

In today's fast-paced and competitive business landscape, organizations are constantly seeking innovative solutions to streamline their operations, accelerate development cycles, and enhance overall efficiency (Lebens et al., 2021). This, however, is a difficult task to fulfil when many businesses cannot find enough professional software developers to develop these innovative solutions (Roslan & Smialek, 2023; Sahay et al., 2020). This shortage of skilled software developers poses a significant challenge to businesses (Lebens et al., 2021; Pham, 2021). To find a viable solution for the shortage of software developers, the cause of the shortage first needs to be understood.

There are multiple potential causes as to why the demand for software developers has increased exponentially in recent years. Firstly, the digitalisation of business processes has become a near necessity in the current highly competitive business environment (European Investment Bank, 2023; Tulchynska et al., 2021). To optimise their operations, increase efficiency and stay ahead of the curve, companies are investing in the digital transformation on a large scale (Breaux & Moritz, 2021; Pham, 2021). This increased reliance on technology and increased complexity translates into a higher demand for software developers who can design, develop, and maintain complex software applications (Bernsteiner et al., 2022; Bock & Frank, 2021; Ozkaya, 2022). Secondly, the rapid evolution of technology has led to a decrease in innovation cycle time (Wen et al., 2022). Emerging technologies such as artificial intelligence (AI), machine learning, blockchain and Internet of Things (IoT) are some of the examples of this. Businesses are keen to harness the potential of these technologies as to gain a competitive edge over their competitors (Wen et al., 2022). The implementation and utilisation of these technologies, however, require skilled software developers, again increasing the demand. Furthermore, the exponential rise of mobile application and a cloud-first approach have further intensified the demand for software developers (Danhieux, 2022). Mobile apps and websites have become an integral part of businesses, and all require skilled software developers, not only for their development, but also for upkeep (Pham, 2021).

Meanwhile, the supply of software developers has not kept pace with this growing demand, resulting in an imbalance (figure 1) (Breaux & Moritz, 2021; Lavelle, 2019; Pham, 2021). There are again, multiple potential reasons for this. Firstly, traditional academic programs can often struggle with keeping up with the rapidly evolving technology landscape, potentially leading to a mismatch in skills acquired by graduates and skills required by businesses (figure 1) (Beręsewicz et al., 2023; Binzer & Winkler, 2023; Raimi, 2021). Furthermore, the competitive job market further exacerbates the shortage, due to all organisations competing for the same pool of skilled software developers (European Investment Bank, 2023). This leads to an increase in salaries and benefits, making it more challenging for smaller organisation to compete and hire the needed talent (European Investment Bank, 2023; Pham, 2021).

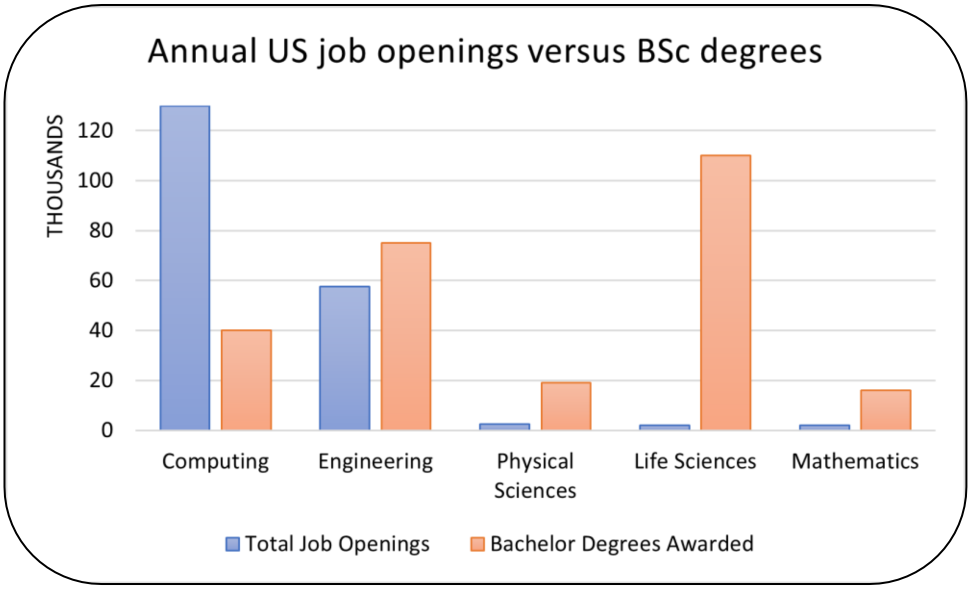


Figure 1 An overview of the annual total US STEM job openings job versus the bachelor's degrees awarded per sector in 2022. The blue represents the total job openings whereas the orange represents the new supply of suitable job candidates. Adapted from Alex Melnichuk, (2022).

This shortage can have far-reaching implications for businesses. It can hamper the organisations' ability to innovate and keep up with their competitors (Raimi, 2021). But it can also have a direct impact on business operations, seeing as many businesses rely on software applications to support their core business processes. According to Lavelle, (2019)**,** the shortage of software developers is the number one risk to organisations globally. Therefore, this shortage has prompted a shift in the way some organisations approach software development, leading to the rise of citizen developers (Abdullah Al Alamin et al., 2021; Lebens et al., 2021).

Citizen developers are employees with domain knowledge, who often lack the formal training in software development, but can make use of the rise in Low-Code No-Code (LCNC) platforms to build business applications themselves (Lebens et al., 2021). These Low-code No-code platforms use drag-and-drop technology with predefined modules to build an application (Hyun, 2019). LCNC platforms provide intuitive interfaces, pre-built components along with a multitude of functionalities, to allow citizen developers to create applications without extensive coding knowledge (Hyun, 2019; Sahay et al., 2020). Because of this, businesses in all segments see citizen developers and LCNC frameworks as a potential solution to the shortage of software developers (Lebens et al., 2021).

Low-code No-code platforms

Low-code No-code platforms have recently emerged as a new class of development environments. The term Low-code No-code platform can be interchangeably used with the terms Low-code platform (LCP) or low-code development platform (LCDP). The term Low-code no-code was presumable coined in 2014 by Forrester, a market research company, in a market analysis report called 'The Forrester Wave' (Bock & Frank, 2021; Richardson et al., 2014). In the report it was stated that LCNC platforms could be an important strategy to deliver software at speed, thereby aiding in the acquisition of customers. A prediction that application development and delivery (AD&D) would embrace and adopt LCNC platforms was also stated, mentioning that as vendors adopt them, LCNC platforms will likely scale up (Richardson et al., 2014).

As of late, LCNC platforms have begun to scale up, as can be seen from the incorporation of LCNC into the product portfolio of large software vendors, such as Microsoft IBM and Oracle. Even non-software vendors, such as Siemens, have stepped into the LCNC market, with the acquisition of Mendix in 2018 (Liu & Zhu, 2022). These movements have led to a self-perpetuating trend with many buzzwords such as Low-code and citizen developer. These trends however do not help simplify the meaning of the platforms or explain the concept of LCNC platforms. There are a few definitions of what LCNC platforms are. According to Vincent et al., (2019), a LCNC platform can be defined as the following:

> "A low-code application platform (LCAP) is an application platform that supports rapid application development, one-step deployment, execution and management using declarative, high-level programming abstractions, such as model-driven and metadata-based programming languages. They support the development of user interfaces, business logic and data services, and improve productivity at the expense of portability across vendors, as compared with conventional application platforms."

In short, a LCNC platform is an application platform where you can create applications while having no real coding knowledge. Most LCNC platforms use drag and drop functionalities instead of writing lines of code. This is also where the term Citizen developer comes into the picture. As mentioned before, citizen developers are employees with domain knowledge, who often lack the formal training in software development. Citizen developers are the focus group of LCNC platforms and are the main reason why LCNC platforms can be an efficient tooling. If people with domain knowledge no longer have to relay that knowledge to professional developers, but can directly create the applications themselves, the efficiency increases. The same can be said for complexity and the chance of errors in business logic. Even though the complexity of the application itself has an upper limit when it is made through a LCNC platform by a citizen developer, the complexity of the business logic is not limited to the knowledge and business capability of the professional developer. In short, the basic idea of LCNC and Citizen Development is to hide the complexity of coding through a high level of abstraction while ensuring full functionality of the code (Carroll et al., 2021; Sahay et al., 2020). This in turn, leads to an enhanced ease-of-use allowing subject matter experts (SME) with minimal or no coding skills to design, develop, and deploy their own, lightweight applications and workflows (Binzer & Winkler, 2023; Carroll & Maher, 2023; Sahay et al., 2020).

Problem definition

There are still hurdles to overcome when it comes to embracing citizen developers and by extension, LCNC frameworks as the solution to the shortage of software developers. Firstly, there are many different LCNC frameworks currently available. Organisations should carefully select and leverage LCNC platforms which align with their specific requirements or technical landscape. Which one to choose may differ for every business. It can, however, be somewhat difficult to assess each framework. Therefore, this comparative study aims to evaluate three prominent LCNC frameworks, namely: SAP Build, Mendix RAD, and Neptune DXP from the perspective of a citizen developer. Furthermore, the study aims to provide insights into these frameworks from the perspective of an SAP environment, where organizations heavily rely on SAP's comprehensive suite of enterprise software solutions.

The primary objective of this comparative study is to assess the strengths and weaknesses of each framework concerning features such as security support, SAP integration, ease of use, scalability, extensibility, and overall development capabilities. According to Bock & Frank (2021), in the world of LCNC, there is a propensity towards market jargon. Therefore, by conducting an in-depth analysis of these platforms from the viewpoint of a citizen developer, businesses can make an informed decision when selecting a LCNC framework.

To do so, this report aims to answer three questions:

1. How do the three LCNC frameworks, SAP Build, Mendix and Neptune, compare?
2. How big / disruptive is the impact of LCNC frameworks in the IT world?
3. How can the strong suits of each framework be leveraged to maximise impact?

Furthermore, two questions related to Accenture’s strategic positioning and its clients will be answered, namely:

4. In what aspects can Accenture clients benefit from using LCNC frameworks?
5. Why and how should Accenture position LCNC frameworks to its clients?

How these questions will be answered, as well as in what manner the LCNC frameworks will be compared is described in the methodology of this report. This includes the definition of ‘impact’ that will be used in this report.

The three Low-Code No-Code platforms

SAP Build

SAP Build is a LCNC platform that allows business users to provide solution by creating applications, automating workflow processes, and designing engaging business sites without writing code. SAP Build is a LCNC platform designed by SAP specifically for SAP environments. The comprehensive set of pre-built components and templates are used to integrate in SAP systems (SAP, n.d.-b). With these integration capabilities, SAP Build aims to streamline the development process by leveraging existing SAP modules, APIs, destinations, and data models. The SAP Build environment runs on the SAP Business Technology Platform and consists of three different parts: SAP Build Apps, SAP Build Process Automation and SAP Build WorkZone (figure 2).

SAP Build Apps

Sap Build Apps, formerly known as SAP AppGyver, is a visual development solution to create enterprise apps without writing code. The primary function of SAP Build Apps is to design mobile applications. This can be done through drag-and-drop functionalities with pre-built components. These components can then be configured to the needs of the user. SAP Build Apps can integrate with backend data, including SAP systems (SAP, n.d.-a).

SAP Build Process Automation

SAP Build Process Automation is a platform which enables the management of automation of workflows and processes using visual features. It is a combination of SAP Workflows Management and SAP Intelligent Robotic Process Automation (RPA). Furthermore, many pre-built workflows are available to the user, which can be adapted to their specific needs (SAP, n.d.-c).

SAP Build WorkZone

SAP Build WorkZone was created from SAP Launchpad and as a subset of SAP SuccessFactors WorkZone. It can be used to create a business site, which serves as a single access point to business applications, processes, and further information. SAP Build WorkZone enables users to build and customize business sites with a drag-and-drop capabilities and enables content creation and page building (SAP, n.d.-d).

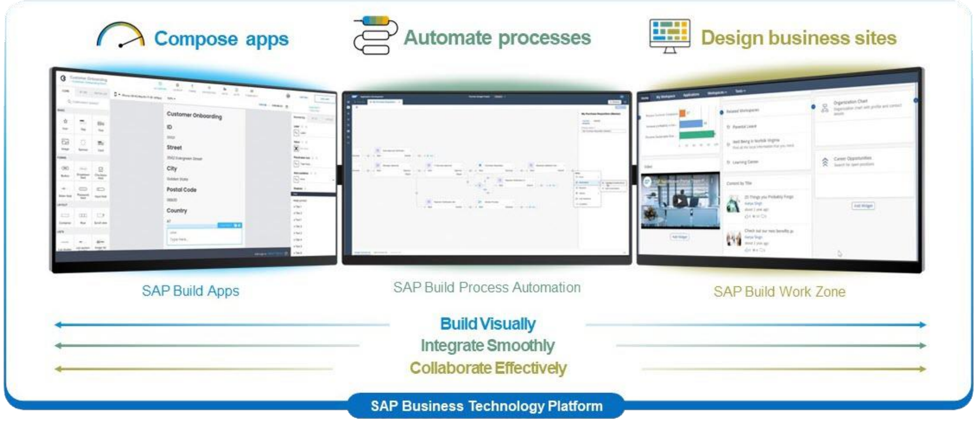


Figure 2 An overview of the structure of the LCNC framework SAP Build. SAP Build Apps, SAP Build Process Automation and SAP Build WorkZone are the three elements of this framework (SAP, n.d.-b).

Mendix

Mendix is a widely used, visual model-driven LCNC platform that does not require any code writing for the development of web-based and mobile applications. All features can be accessed through drag-and-drop capabilities while collaborating in real-time with peers and colleagues. Mendix aims to increase the efficiency of workflows and processes for business applications, as well as help accelerate the entire lifecycle development of apps all through forming a base for collaboration (Marc & Mileff, 2023). For this purpose, Mendix used to offer two integrated development environments (IDE), namely the web-based Mendix studio and the desktop-based Mendix studio Pro (Gurcan & Taentzer, 2021).

Mendix Studio

A web-based development environment aimed towards the no-code approach of creating applications. Mendix studio was mainly focused on business developers with primarily pre-programmed elements and drag-and-drop functionalities.

Mendix Studio Pro

Mendix Studio Pro was the desktop-based development environment used primarily by more experienced developers. The same type of applications could be developed, with the possibility of making additional adjustments through JavaScript (Gurcan & Taentzer, 2021).

These two approaches together made it possible for citizen developers and professional developers to create applications separately or combined, thereby increasing the exchange between the domain knowledge of citizen developers and the expertise of the professional developers. Recently, these two environments were merged into one environment Mendix 10, according to Mendix to bridge the gap between IT and business even more (de Visser, 2023). Mendix is seen as a leader by both the Forrester wave evaluation, as well as the Gartner Magic quadrant (Bratincevic et al., 2023; Vincent et al., 2022).

Neptune

Neptune DXP (Digital Experience Platform) is a relatively unknown LCNC framework designed to enhance SAP user experience and extend SAP applications. It provides a comprehensive set of tools and functionalities to create modern and responsive user interfaces, incorporating SAP's Fiori design principles. Neptune DXP aims to provide seamless integration with SAP backend systems, allowing developers to leverage existing SAP functionalities. Neptune has two different versions available, namely SAP Edition and Open edition. The SAP edition was developed to run embedded within the SAP S/4HANA or the SAP Business Suite and is focused on ABAP developers (Neptune, n.d.-b). The Open edition is not limited to the SAP run environment but instead focused on a more open-standard based platform (Neptune, n.d.-a).

A potential solution

LCNC platforms have the potential to be (part of) the solution to the limited supply of professional developers. Furthermore, the increased rate of innovation and by extension, the cost increase to keep up, can potentially be accounted for by utilising LCNC platforms. All LCNC frameworks, if utilised and implemented correctly, have the potential to accelerate time-to-market, especially in an already agile business environment (Lebens & Finnegan, 2021).

Approach

This is a study done from the perspective of a 'new' citizen developer (the author of this report). Therefore, all three LCNC frameworks in this study are unfamiliar to the author and all experiences during the duration of this study will be used to answer the aforementioned research questions. The findings of this study will therefore be (partially) reliant on the tutorials and guides of the three LCNC frameworks available and also be influenced by the digital environment preferences of the author. Since the study is done from the perspective of someone who is not a professional developer, the depth of the analysis on the more technical parts of this study can potentially be lacking for those with more technical knowledge.

While reading this study it is good to keep in mind that LCNC and Citizen development are practice driven phenomena that are closely intertwined. Therefore, stating that anyone can use LCNC is not necessarily true. There needs to be a certain aptitude for it, as well as a lot of practice to make it efficient to use (Binzer & Winkler, 2023).

Additionally, a part of this study is entirely done from the perspective of SAP environment, meaning the features and the strengths and weaknesses of SAP Build will be based on the usage of SAP Build within an SAP environment.

Methodology

Comparison method

To assess how the three LCNC frameworks, SAP Build, Mendix and Neptune compare, first a method of comparison needs to be selected. In this report, a list of features will be drawn up on which to compare the three different platforms. Then the features will be compared between the three platforms in a three-step process.

1. The features list is divided into two sections. The first section will only contain the 'mandatory' features, which make a LCNC platform a LCNC platform. Overall, mandatory features are more of a feature group that needs to be present in some way or form. There can still be parts of the feature group missing, but the essence of the feature group needs to be present. E.g., every LCNC framework needs to have a Graphical User Interface, but the sub-features of this feature group, e.g., 'pre-built forms' or 'progress tracking' are not necessarily available. The second section will contain these 'nice-to-have' features, focused on usability and efficiency. These features can be gathered from literature and through empirical research.
2. The second step in this process focuses on the quantification of the qualitative data, to give a value to a result with which to compare. First, whether each mandatory feature is present at the specific LCNC framework will be noted and the way the feature is implemented will be assessed. The mandatory features, as the name implies, need to be present for a LCNC framework to be considered a true LCNC framework. The mandatory features used in this report are derived from Roslan & Smialek (2023) and Sahay et al., (2020) . The sub-features will then be assessed and will be given a score. The scoring for the features will be subjected to are as shown in table 1. The sub-feature is either absent, implemented, or optimised. An absent feature will receive no score since the feature is not present and can therefore not be tested. An implemented feature is a feature assessed to be present but lacking in some ways, e.g., not intuitive to use or does not work in all situations it should. A score of 1 is given to an implemented but not optimised feature. An optimised feature works intuitively and in all situations. No roundabout method is needed to use the feature to its full capability. A score of 2 is given to an optimised feature.
    a. Some features on which the LCNC framework will be compared will not fit into the same scoring terms. These features are mostly comprised of parameters which are not integrations or functions, but rather different comparison parameters. E.g., features such as 'Cost' or 'Ease of use'. The same scoring values and grading will apply to these features; however, the terms will not apply correctly. For these features the grading of the score will from lowest to highest, meaning a score of 0 will be the worst scoring and a score of 2 will be the best scoring.

Table 1 The scores given to a certain degree of implementation of the features of a LCNC framework as given in this study. A missing sub-feature is given a score of 0. Implemented but not optimised is 1 point. Optimised features are given 2 points.

Feature	Score
Absent	0
Implemented	1
Optimised	2

3. Lastly, the results of the comparison will be summed, and the platform with the most points has the overall highest potential or is the most well-rounded. The sub-sets of features and the scores can be used further in this report to assess the strengths and weaknesses of the LCNC frameworks in conjunction with empirical research. Since this report is from the perspective of an SAP environment, the focus of this quantification of the features and subsequent analysis and discussion is to determine in what way the SAP Build framework differs from the Neptune and Mendix frameworks.

Furthermore, a more in-depth strength and weakness assessment will be made possible by comparing the total score per group of features per LCNC framework as well. By comparing the group of features score between the three different LCNC frameworks, the strong suits and weaknesses of the LCNC framework will be made clear.

Empirical research

The sub-features of these feature groups are also assessed through empirical research. This research is done by familiarising with each of the three LCNC frameworks through their subsequent beginner tutorial guides. After these beginner tutorials, some intermediate guides will be followed to aid in a demo app creation for each of the frameworks. The type of demo applications made will be based on the perceived strong suits of each LCNC framework. Some of the demo applications used will potentially follow a guide, if available. The perceived strong suits will be based on literary research of the three LCNC frameworks.

The empirical research will especially aid in the comparison of the perceived strengths and weaknesses of the different LCNC frameworks since a quantitative method as described above scores every feature with the same weight. The weight of each feature can be different depending on how good the actual feature is implemented and optimised, but also by how large a part it can play in the LCNC decision of businesses. For example, the Process management feature group is scored the same as the GUI feature group, however, for some businesses the function of the Process management feature group of a LCNC framework outweighs the form and function of the GUI feature group.

Impact

After the quantitative and qualitative comparison methods are completed, along with the assessment of the strengths and weaknesses of the three LCNC frameworks, the impact LCNC in general has made on the IT world is studied. The impact LCNC has already made and is forecasted to make will help in assessing the worth of LCNC frameworks for businesses. If possible, the impact LCNC already has on specific sectors will be assessed and this impact will be cross examined with the strength and weaknesses of the three LCNC frameworks to determine in what sector specifically each LCNC framework will come best to fruition. Furthermore, when the strong suits are known, they can be analysed to determine how to leverage each framework to maximise its impact. This will potentially aid in the recommendation of a LCNC framework for a specific client for Accenture in the recommendation part of this report.

The impact of LCNC on the IT world is difficult to measure. First, what counts as impact? For this report, impact will be determined by the general increase of usage of LCNC frameworks throughout the IT world. In addition to this, the effect this usage has had on the businesses themselves will be assessed. This can be in the form of savings on development, speed of development or time-to-market, or even on perceived workload.

To determine the effect and impact LCNC has already had and will potentially have on the IT world, surveys of IT professionals are needed. However, conducting surveys of this scale are not in the scope of this study. Therefore, surveys and studies which are already conducted and analysed will be used. These surveys will be gathered from academic and grey literature. Mainly surveys which relate to the actual use or increase in usage of LCNC frameworks within companies will be looked at. Furthermore, the effects of the implementation (if applicable) of LCNC frameworks within the portfolios of these companies, such as effect on development speed or budget, will be assessed.

In addition to this, my own experiences with citizen development as well as interviews with starting citizen developers can help give insight in the process of integrating LCNC frameworks into a pre-existing IT landscape within a company. These insights can also help paint a picture as to how low-code development can have an impact on the individual within a company.

All the findings through these methods can assist in formulating how Accenture and its clients can benefit from using LCNC frameworks and how Accenture should position LCNC frameworks to its clients.

Definition of features

The features which will be compared between SAP Build, Mendix and Neptune, must first be defined. The following list of feature groups will be used to give a broad overview of how each of the three LCNC platforms score on the basic features (Khorram et al., 2020; Roslan & Smialek, 2023; Sahay et al., 2020). The specific sub-features per feature group will be specified in the results of this report, before a score is given.

These are the overall base feature group on which the three LCNC frameworks will be compared:

- Graphical user interface (GUI)
  - The GUI feature group consists of front-end functionalities which enable interactions with the user. E.g., drag-and-drop tools, forms, or progress tracking are sub-features which belong to this feature group. The ease of use of a LCNC platform is highly influenced by how well implemented the sub-features of this feature group are.

- Interoperability support with external services and data sources
  - The interoperability support feature group relates to the options of interaction with external services. For example, the interoperability with services such as Google drive or SharePoint is considered, but also the integration or interaction with SAP systems. Furthermore, the ability to design forms or reports through the connection with different data sources is also assessed.

- Security support
  - The security support feature group relates to all security related aspects that are considered when an application is developed through a LCNC platform. Features such as User access control or Authentication methods belong to this feature group.

- Collaborative development support
  - This group of features is related to how collaboration of application development is handled, either online or offline. Be it on-premises or cloud collaboration support.

- Reusability support
  - This related to the pre-built or pre-defined content such as templates or dashboards, but also whether the previously created content by the user can be reused.

- Scalability support
  - Can the application be scaled up according to different parameters, such as data traffic, number of manageable active users, storage capabilities.

- Business logic specification mechanisms
  - The Business Logic specification feature group relates to the possibilities on how to specify business logic in the application itself. Sub-features of this include graphical workflow editors or business rules engines themselves. A large part of this feature group relates to API support for communication between applications and how business logic handles and implements API calls. For SAP Build specifically, destinations relate to this feature group.

- Maintenance support
    - The features related to this feature group determine how well a LCNC framework can be maintained after or during deployment, as well as during the development of an application or process.

- Application build mechanisms
    - The Application build mechanisms feature group relates to the technical manner in which an application is built. Whether the LCNC process or application developed does so through employing code generation techniques or whether it utilises an approach of compiling through models at run-time.

- Deployment support
    - The Deployment support feature group relates to the actions of deploying the application. And example of sub-features of this group are concerned with whether the LCNC framework supports cloud deployment or if the application can be deployed on every type of on-premises infrastructure. Deployment support feature group relates to how an application can be published as well as to what it can be published.

- Kinds of supported functions
    - The features in this feature group are top-level features or functions of a LCNC framework. For example, event monitoring or approval processes.

- Accessibility
    - This relates to how accessible a LCNC framework is. This can pertain to subjects such as costs, but also to the ease of use or entry barrier for new users and citizen developers.

Demo Applications

Three separate demo applications will be made for the three different LCNC frameworks. These demo applications will be made after a beginner's tutorial is followed of the respective LCNC framework. The demo applications will aim to be focused on the strong suits of each LCNC framework, based on the limited perspective at the time.

SAP Build

The demo application made on the SAP Build framework will be focused on the perceived strength of the SAP Build environment, namely a process built using the SAP Build Process Automation platform, with the integration of an application built with the SAP Build Apps platform, which utilises the process. The application shows the integrative capabilities of the SAP Build environment.

Mendix

The demo application made with the Mendix framework will be focused on the well-balanced capabilities of Mendix studio. Therefore, a demo application will be made which makes use of the business rules engine as well as some process management and UI design tools. Furthermore, the authentication and authorisation mechanisms within the Mendix Studio environment will be tested, along with the collaborative nature of the web-based Mendix portal.

Neptune

The strong UI/UX capabilities of Neptune will be utilised to create a dashboard for the New Joiner Experience (NJX) of Accenture. The pre-built components of Neptune, along with the modern design of said components make it suitable for such an application. The application will be designed to aid new joiners of Accenture through their first week, with navigation and timetable present.

Results

Feature lists

The feature list consisting of the feature groups and the sub-features on the basis of which SAP build, Mendix and Neptune will be compared is gathered from literary research and empirical research. These features are in part derived from reports written by Abdullah Al Alamin et al., (2021), Sahay et al., (2020) and (Roslan & Smialek, 2023). The provided features from the aforementioned reports have been expanded upon to include non-implementable features such as cost or ease-of-use.

Graphical user interface

- Drag-and-drop designer
  The main GUI feature of LCNC frameworks is the drag-and-drop functionality. This feature is focused on the user experience and enables a user to visually construct an application, including connections, actions, and responses.

- Point and click designer
  The point-and-click approach is a similar approach to drag-and-drop. It is also a visual method of designing the applications, except it involves point to an object and clicking on the interface, rather than drag-and-drop functionalities.

- Forms & reports (pre-built)
  Pre-built forms and reports are pre-built functionalities of a LCNC framework, where a form or report does not have to be manually constructed. It is a reusable form or report with basic functions which a (citizen) developer can utilise during the development of an application.

- Dashboards (pre-built)
  Pre-built dashboards are also a reusable function that enables a user to skip designing a dashboard themselves.

- Forms
  Form can assist with improving the user experience and user interface when creating applications. Forms can include a plethora of options, such as surveys, checklists, or dashboards, many of which can improve the usability of an app.

- Progress tracking
  Progress tracking enables multiple people to work on an application at the same time. The progress of the application development can be tracked through this feature and helps collaborators to work simultaneously or after one another on the same application.

- Advanced Reporting
  Advanced reporting is a feature which enables to user to get a valuable insight into the application usage. This is commonly done through graphs and charts, also known as graphical reporting. Advanced reporting is the ordering and breakdown of data and information in a way in which it makes sense to the user.

- Workflows (built-in)
  Common workflows can be built into LCNC frameworks, seeing as they are often used by users during the development of an application.

- Configurable workflows
  Configurable workflows are basic workflows which can be changed to fit the users needs. Configurable workflows enable a user to customise an application to gather and process the relevant information and data needed.

Interoperability support

- External services interoperability
  The external services interoperability feature encompasses the manner in which platforms or services such as Google, Microsoft or OpenAI is handled. This includes the integration or interoperability with SAP systems.
- Data sources connection
  The data sources connection feature determines the connection of the developed application with relational databases such as Azure or Microsoft SQL or non-relation databases such as MongoDB.
- Internal services (Destinations)
  The interoperability with internal services relates to the way in which the LCNC framework handles endpoints. E.g., the way sensitive information is stored and used in an application.
- SAP Integration
  SAP integration is related to how the different LCNC frameworks handle the integration into the SAP environment S4/HANA. This can mean that the SAP Fiori guidelines are followed or even if and how the LCNC framework can be embedded in the SAP environment.

Security Support

- Application security
  Application security deals with weaknesses with the application itself and determines who can have access to what part of the application depending on the role of the user assign. This feature deals with unauthorised access.
- Platform security
  The platform security deals with the development part of the application security. Who can access and change the application currently being developed. This is most commonly done through role management and authentication mechanisms at the platform level.

Collaborative development support

- Offline collaboration
  Offline collaboration feature enables developers to work on the application offline locally and enables them to commit the changes at a later date. The feature deals with merging of the changes.
- Online collaboration
  This sub-feature deals with online collaboration during the development of the application. Different developers concurrently collaborate on the development and conflicts are handled by the LCNC platform at run-time.

Reusability support

- Built-in workflows
  This feature enables users to reuse workflows if they are commonly used during the development of applications. The workflows can be pre-built by the LCNC framework or by the user at an earlier date.

- Pre-built forms & reports
  This feature enables users to reuse forms and reports if they are commonly used during the development of applications. The forms and reports can be pre-built by the LCNC framework or by the user at an earlier date.

- Pre-built dashboards
  This feature enables users to reuse dashboards or part thereof if they are commonly used during the development of applications. The dashboards can be pre-built by the LCNC framework or by the user at an earlier date. Reusability is often implemented through a 'marketplace', where multiple elements can be uploaded to or downloaded from.

Scalability

- Scalability (users)
  The scalability of the application in relation to users allows the application to increase the capacity in handling more active users at the same time. This feature enables developed applications to have a way array of uses, not limited to the active users.

- Scalability (data)
  The sub-feature of an application to scale-up based on the data traffic volume can be essential if an application is expected to handle a varying degree of requests.

- Scalability (storage)
  The sub-feature for an application to scale the allotted data storage capacity. This feature enables an application to expand or decrease the data storage capability it is connect to.

Business logic specifications

- Business rules engine
  A business rules engine allows the application to execute business rules depending on the user's requirements. Business rules can be dynamically adapted to the provided data.

- Graphical workflow editor
  Business rules applied in a graphical workflow editor enable the developer to connect different business rules to a multitude of scenarios in a graphical manner.

- AI business logic
  The AI enabled business logic feature is a self-learning tool which can help the developer to easily manage and replicate behaviours according to specific, learned mechanisms.

Maintenance support

- Quality control
  Quality control features come in many different forms. Quality control enables developers to determine where improvements are needed depending on different parameters. E.g., heatmaps of user inputs can determine where users struggle with the use of the application.

- Quality management
  Quality management features come in many different forms. The sub-features allow the user to manage and maintain the quality of the application or workflows. If certain errors occur, the quality management features can be used to resolve problems remotely.

- Ease of maintenance
  Ease of maintenance relates to the manner in which maintenance is performed. Whether issues or inefficiencies can be checked through a separate application, tab or if issues or bug fixes need to be resolved manually.

- Error reporting
  The error reporting relates to how the errors which occur during the development process of an application or workflow is handled. This can relate to the way in which the error is made clear to the developer. E.g., Error console with clear language and direction.

- Extensibility
  This function relates to the measure in which a LCNC framework is able to use third party extensions, and the effort required to do so. Extensibility or openness of a LCNC framework can be of use for a wide variety of purposes, such as management tools or testing applications.

Application build mechanisms

- Code generation
  Code generation relates to the manner in which an application is built. Utilising code generation, the source code of an application is generated from specified models.

- Models at run-time
  If the application developed through the LCNC framework is modelled at run-time, the run-time execution is done through an interpretation of specified models for the application.

Deployment support

- Cloud deployment
  This sub-feature enables an application built through a LCNC framework to be deployed straight to a cloud infrastructure.

- Local deployment
  If the specified LCNC framework allows for local deployment, the developed application can be deployed on-premises on the infrastructure of the users' organisation.

- Testing
  This sub-feature is used to assess how streamlined the testing process of the application is. This includes whether the testing is possible on mobile and browsers.

Supported types of functions

- Event monitoring
  Event monitoring enables the developer of the application to analyse the data of a specific event and to determine what caused the event to occur.

- Process automation
  Process automation enables a developer to automate processes or workflows which can be complex. Depending on the specifications of these processes, intervention on part of the user is not necessary.

- Approval process control
  Approval process control is part of the process automation process. This feature enables decision processes to automatically be approved, declined, or sent to the correct authorised user to make the decision.

- Inventory management
  Inventory management is based on an automatic balancing feature, which can help balance the outflow and inflow of goods, information, or finances. This feature enables automation and workflows to operate under dynamic limits.

Accessibility

- User entry barrier
  This feature relates to pre-requisite knowledge a user needs to have to be able to use a given platform. This can be knowledge of different tools and technologies, or general technical, technological, or digital knowledge. Furthermore, the setup process before the LCNC framework can be used for development is considered.

- Documentation availability
  The availability or accessibility of documentation on given features or function within a LCNC framework can help streamline the process of development for a developer. The quality, availability, and accessibility of the documentation in question is what this feature relates to.

- Learning materials
  The learning materials, tutorials or guides which are provided by the platform when first encountering the LCNC framework. The quality of the beginner, intermediate and advanced guides can help reduce the time it takes for a citizen developer to grasp the functionality of a LCNC framework. What can or cannot be done, best practises, as well as how functions are used.

- Community support
  The community of a LCNC framework can help in answering questions that may arise during the development process. A strong community around a LCNC framework can help in pointing towards documentation, directly help with solving (irregular) issues and also provide custom solutions or other ways to tackle problems.

- Learning curve
  The difference in difficulty between the beginner, intermediate and advanced courses of a LCNC guide can give indication towards the learning curve of a LCNC platform. How difficult is it to develop more complex applications or workflows.

- Cost
  The cost of the LCNC framework can play a large role in whether or not a business or developer chooses to use said platform. The cost between the three LCNC frameworks is determined by taking the costs for a single developer to use the platform. A free-to-use version of the framework is seen as a pre-requisite for scoring 2 points on this feature.

Results of the quantification of the data

The results of the quantification of the data are shown in table 2.1 and table 2.2. The features and whether they are present in the three different LCNC frameworks are assessed based on literary research, the product pages of the LCNC framework in question, as well as empirical research and interviews (*Appendix*). As stated in the Methodology, a score of 0 means the feature in question is either absent or very poorly executed. A score of 1 means the feature is implemented but lacking in some ways. A score of 2 means the feature is implemented and works correctly and smoothly.

Table 2.1 Results of the quantification of the features of the three different LCNC frameworks SAP Build, Mendix and Neptune. Score of 0 means absent, score of 1 means implemented but not optimised. A score of 2 is an optimised feature, implemented correctly.

Feature	SAP Build	Mendix	Neptune
Graphical user interface			
Drag-and-drop designer	2	2	2
Point and Click approach	1	1	2
Pre-built forms	2	2	2
Pre-built reports	1	2	1
Pre-built dashboards	0	1	2
Forms	2	2	2
Progress tracking	0	2	1
Advanced reporting	0	2	1
Built-in workflows	2	2	1
Configurable workflows	2	2	1
Interoperability support			
With external service	2	2	2
Connection with data sources	2	2	1
Internal services (Destinations)	2	1	1
SAP Integration	2	1	1
Security support			
Application security	2	2	2
Platform security	2	2	2
Collaborative development support			
Offline collaboration	0	2	0
Online collaboration	1	2	2
Reusability support			
Built-in workflows	2	2	2
Pre-built forms and reports	1	2	2
Pre-built dashboards	0	1	2
Scalability			
Scalability on number of users	2	2	2
Scalability on data traffic	2	2	2
Scalability on data storage	2	2	1

Table 2.2 The second part of the results of the quantification of the features of the three different LCNC frameworks SAP Build, Mendix and Neptune. Score of 0 means absent, score of 1 means implemented but not optimised. A score of 2 is an optimised feature, implemented correctly.

Business logic specification mechanisms			
Business rules engine	2	2	2
Graphical workflow editor	2	2	1
AI enabled business logic	0	2	2
Maintenance support			
Quality control	1	2	2
Quality management	2	2	1
Ease of maintenance	1	2	1
Error reporting	1	2	1
Extensibility	1	2	0
Application Build mechanisms			
Code generation	0	0	0
Models at run-time	2	2	2
Deployment support			
Deployment on cloud	2	2	2
Deployment on local infrastructures	2	2	2
Testing	1	1	1
Kinds of supported functions			
Event monitoring	1	2	1
Process automation	2	1	1
Approval process control	2	2	0
Inventory management	2	2	1
Accessibility			
User entry barrier	0	2	1
Documentation availability	1	2	1
Learning materials	1	2	1
Community support	2	2	1
Learning curve	1	1	1
Cost	1	1	2

The results when comparing the three LCNC frameworks through the scoring method are shown in table 3. Mendix has the highest total score of 82. SAP Build has 63 and Neptune the lowest with 62. The LCNC frameworks scored these values out a maximum total of 92. This translates into Mendix having 89% of the total score possible, SAP Build and Neptune 68% and 67% respectively.

Table 3 The results of the scoring method for the LCNC frameworks SAP Build, Mendix and Neptune. Maximum score possible: 92. The percentage each LCNC framework scored of the maximum score available is also given.

	SAP Build	**Mendix**	**Neptune**
Score	**63**	**82**	**62**
Score (%)	**68**	**89**	**67**

Comparison results

The results for the qualitative comparison method can also be divided by feature groups. This division can give insight as to where the differences in score between the three LCNC frameworks comes from. The score per feature group per LCNC framework can be seen in table 4.

Table 4 The results of the scoring method per feature group for SAP Build, Mendix and Neptune. The maximum possible score per feature group is given in the last column of the table.

Feature	SAP Build	Mendix	Neptune	Maximum
Graphical user interface	12	18	15	20
Interoperability support	8	6	5	8
Security support	4	4	4	4
Collaborative development support	1	4	2	4
Reusability support	3	5	6	6
Scalability	6	6	5	6
Business logic specification mechanisms	4	6	5	6
Maintenance support	6	10	5	10
Application Build mechanisms	2	2	2	4
Deployment support	5	5	5	6
Kinds of supported functions	7	7	3	8
Accessibility	5	9	5	10

To visualise this data, the comparison of the score per feature group per LCNC framework is also shown in the figure 3 for the absolute score comparison. The score per feature group as a percentage of the maximum possible score of that feature group is also shown in figure 4. This enables assessment of these LCNC frameworks to not only be in comparison to the other LCNC frameworks in this study, but to also assess them on their potential in the LCNC world itself.

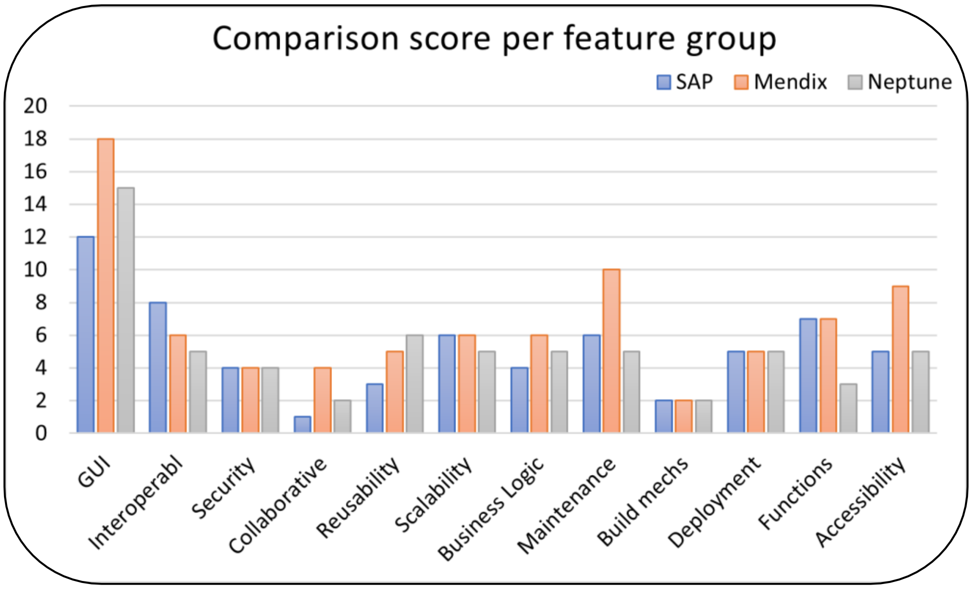


Figure 3 The score comparison per LCNC framework per feature group are shown in this figure. The absolute scores of each feature group are used.

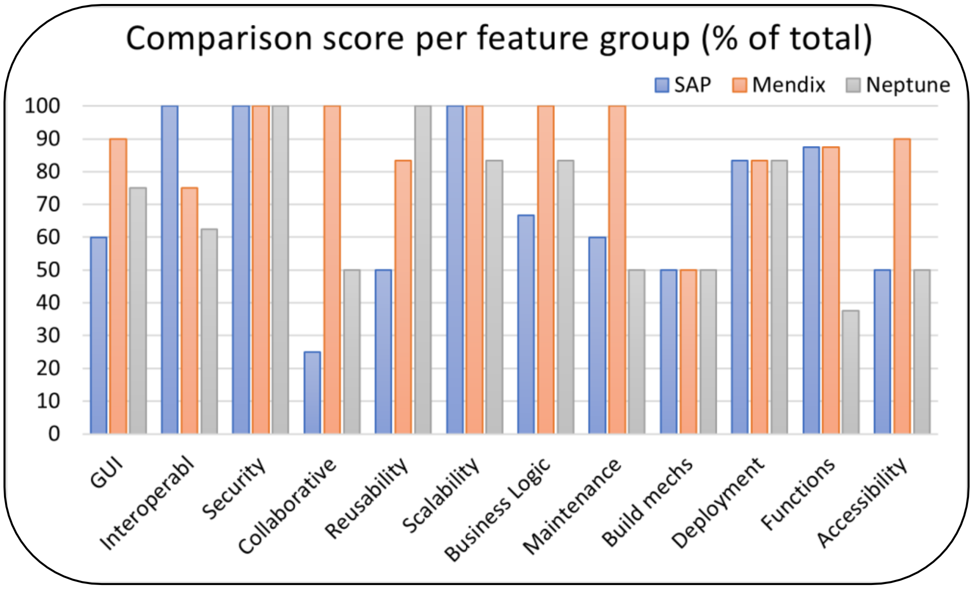


Figure 4 The score comparison per LCNC framework per feature group as a percentage of the total possible score per feature group are shown in this figure. The percentage scores of each feature group are used.

The comparison results per feature group shows large outliers in the ‘Maintenance support’ feature group and ‘Accessibility’ feature group in absolute values. In these feature groups Mendix scored much higher relative to the maximum score possible than SAP Build and Neptune did, with respectively 10 out of 10 and 9 out of 10 for ‘Maintenance support’ and ‘Accessibility’.

Furthermore, an outlier in the low end can be found in the ‘Collaborative development support’ feature group. Here the LCNC framework SAP Build scored 1 out of 4. This is only a quarter (25%) of the maximum available points. Another outlier on the low end can be seen in the ‘Kinds of supported functions’ feature group. Here, Neptune scored a total of 3 out of the maximum of 8. This is approximately 38% of the maximum available points.

With ‘Application build mechanisms’ taken out of the equation (always a score of 50% of the maximum, seeing as the sub-features is either the one or the other), Mendix scores an average of 90% of the maximum score available. SAP Build and Neptune scored a respectively 71% and 70% of the maximum score available.

Strengths and Weaknesses

The strengths of a LCNC framework as per the comparison results are defined as scoring higher than the other LCNC frameworks in this study. Consequently, the weaknesses are defined as scoring lower than the other LCNC frameworks in this study. Relative strengths and weaknesses can be determined by assessing at which feature group the LCNC framework is scoring lower than its average score and where the LCNC in question is outperformed by another LCNC framework. To visualise the differences in score per feature group per LCNC framework, a radar chart is provided (figure 5).

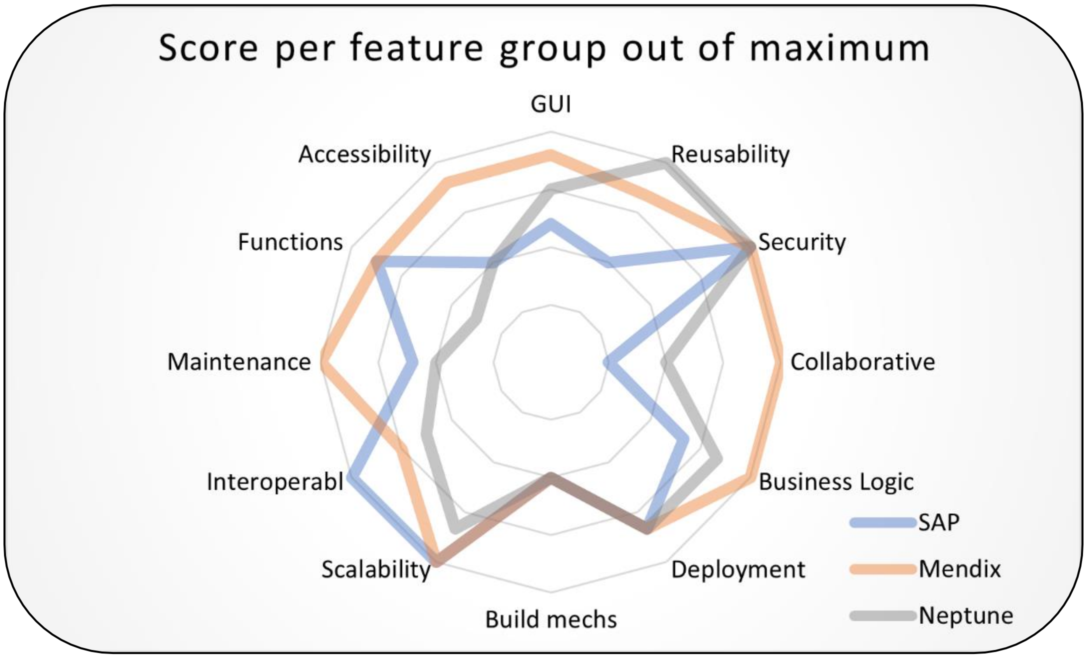


Figure 5 The score per feature group per LCNC framework are visualised. The scores given to the feature groups per LCNC framework are translated to the ratio to the maximum for that feature group.

SAP Build

The strengths of the LCNC framework SAP Build according to the quantitative comparison results is the feature group 'Interoperability support'. SAP build scored the highest out of the three tested LCNC frameworks in this feature group with score of 8 out of 8 (table 4). Furthermore, a relative strength of SAP Build is the feature group 'Kinds of supported functions', where it tied with Mendix in score, but outperformed Neptune (figure 5).

The weaknesses of SAP Build according to the quantitative comparison results are the feature groups 'Collaborative development support' and 'Reusability support'. In these feature groups SAP Build scored 1 out of 4 points and 3 out of 6 respectively. Furthermore, a relative weakness of SAP Build is found in the feature group 'Accessibility', where it tied with Neptune in score, but is significantly outperformed by Mendix (figure 5).

Mendix

The strengths of the LCNC framework Mendix according to the quantitative comparison results are the feature groups 'Graphical user interface', 'Collaborative support', 'Business logic specification mechanisms', 'Maintenance support', and 'Accessibility'. Mendix scored the highest of the three LCNC frameworks in all of these feature groups (table 4). For the feature groups 'Collaborative development', 'Business logic specification mechanisms' and 'Maintenance support' Mendix scored the maximum possible score. For the feature groups 'Graphical user interface' and 'Accessibility' Mendix scored 18 out of 20 and 9 out of 10 respectively. The relative strengths of Mendix are the feature groups 'Scalability' and 'Kinds of supported functions'. In both of these feature groups Mendix scored the same as SAP Build. In both cases this was the maximum score (figure 5).

According to the quantitative comparison results Mendix does not have any clear weaknesses when assessing the feature groups themselves (table 4). However, two relative weaknesses of Mendix can be found. The feature group 'Interoperability support' is a relative weakness of Mendix when compared to SAP Build. Mendix got a score of 6 out of 8, whereas SAP Build got the maximum score for this feature group. Furthermore, in the feature group 'Reusability', Neptune had a score of 6 out of 6, whereas Mendix score had a score of 5 (figure 5).

Neptune

The strengths of the LCNC framework Neptune according to the quantitative comparison results is the feature group 'Reusability support'. Neptune scored the highest out of the three tested LCNC frameworks in this feature group with a maximum score of 6 out of 6 (table 4). Furthermore, there are two relative strengths of Neptune when compared to the other LCNC frameworks. These two relative strengths are the feature groups 'Graphical user interface' and 'Business logic specification mechanisms'. For the feature group 'Graphical user interface', Neptune scored 15 out of 20, whereas Mendix score 18 out of 20 and SAP Build scored 12 out of 20. Neptune received a score of 5 out of 6 for the feature group 'Business logic specification mechanisms', only outperformed by Mendix with a maximum score of 6 out of 6.

The weaknesses of Neptune according to the quantitative comparison results are the feature groups 'Kinds of supported functions', 'Maintenance support' and 'Scalability' (table 4). In the first feature group Neptune scored 3 out of a maximum of 8, whereas SAP build and Mendix both scored 7 out of 8. In the feature group 'Maintenance support' Neptune received a score of 5 out of 10. In the feature group 'Scalability', Neptune received a score of 5 out of 6, whereas the other LCNC frameworks in this study received the maximum score of 6. Furthermore, a relative weakness of Neptune is found in the feature group 'Accessibility', where it tied with SAP Build in score, but is significantly outperformed by Mendix (figure 5).

Empirical Research – Demo Applications

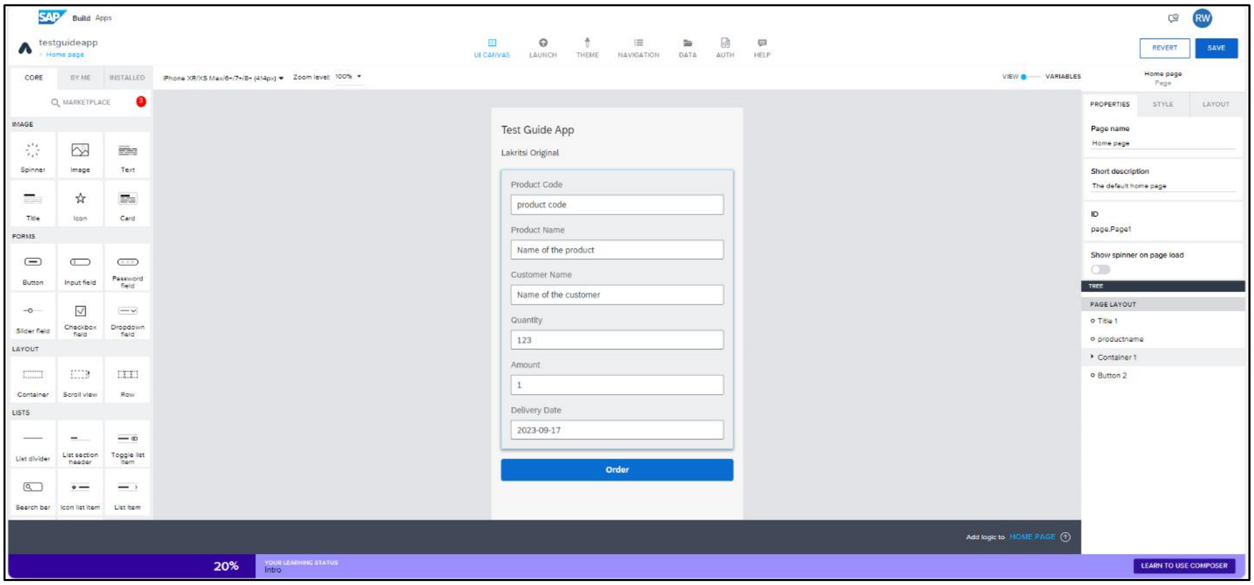
Qualitative

The qualitative results, resulting from the empirical research of the three different LCNC frameworks SAP Build, Mendix and Neptune are difficult to compare. However, the weaknesses and strong suits as experienced by a citizen developer (the writer of this report) of each framework are noted down. The comparison is therefore based on the perceived usability of each framework for certain functions and not necessarily the actual limits of the framework itself.

SAP Build

For SAP Build, the empirical research consisted of a demo application made using primarily SAP Build Process Automation and SAP Build Apps. The process created is focused on the approval of sales orders. These sales order need to be approved automatically depending on a threshold of quantity or value. If the quantity or value of the order surpasses the threshold, the approval of said order needs to be directed to the correct person in charge to be manually approved or declined. The demo application itself was made using SAP Build Apps. The function of this application is to initiate sales orders sent to the sales order process. This demo application is in part made during an intermediate level guide of SAP Build. The goal of the application is to show the integrative capabilities of the SAP Build environment. SAP Build WorkZone was not used to integrate the application and process in a business website, since the trial version of SAP Build does not include the full required capabilities of SAP Build WorkZone.

The SAP Build Apps UI canvas is shown in figure 6. This figure shows the application as seen from the developers' point of view. This where all most of the drag-and-drop functionality of the SAP Build Apps platform takes place.



Test Guide App

Lakritsi Original

Product Code: product code

Product Name: Name of the product

Customer Name: Name of the customer

Quantity: 123

Amount: 1

Delivery Date: 2023-09-17

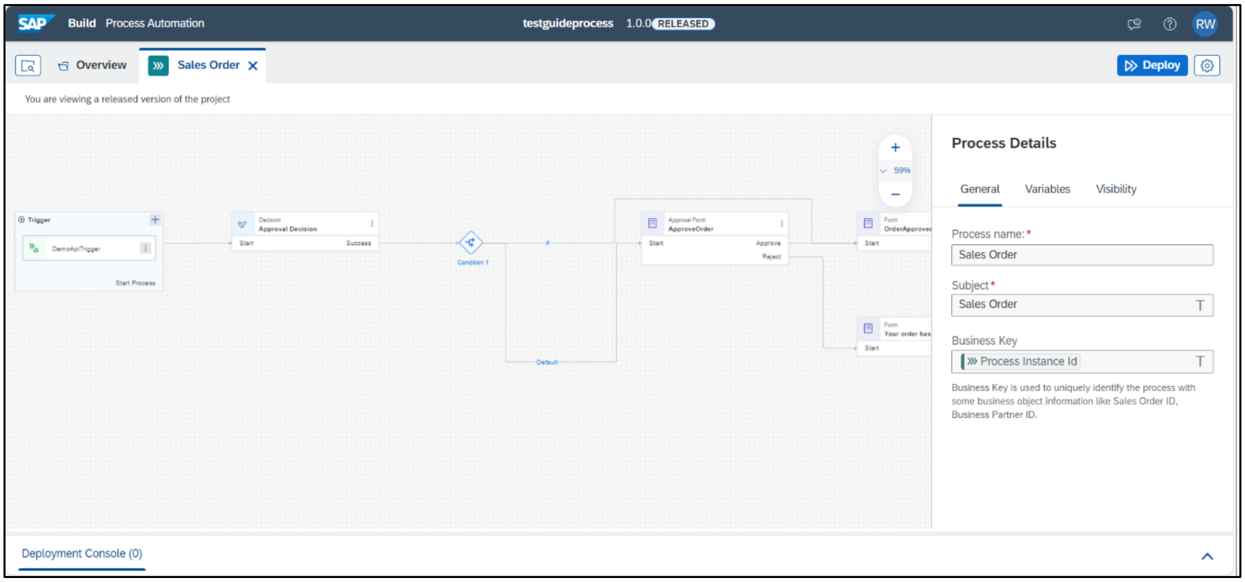
Page name: Home page

Short description: The default home page

ID: page.Page1

Figure 6 The main UI canvas of SAP Build Apps is shown in this figure. The drag-and-drop components are on the left-hand side of the canvas, the logic can be added to the components at the bottom grey bar. The tabs at the top of the screen open many of the main functionalities, such as deployment, data connections and authentication mechanisms.

The SAP Build Process Automations process page can be seen in figure 7. Here the business logic of the process is shown. The process works from left to right, where on the left-hand side the trigger is located, which is the point of entry for sales orders created through the SAP Build Apps application shown in figure 6. Each tile in the process signifies a different function and a step in the process as a whole.



You are viewing a released version of the project

Process name: * Sales Order

Subject * Sales Order

Business Key Process Instance Id

Business Key is used to uniquely identify the process with some business object information like Sales Order ID, Business Partner ID.

Figure 7 The demo process made with SAP Build Process Automation is shown in this figure. Trigger on the left-hand side of the canvas is the point of entry for the sales orders, where it follows the process step-by-step as seen in the figure. Each step can be configured according to the developer’s needs.

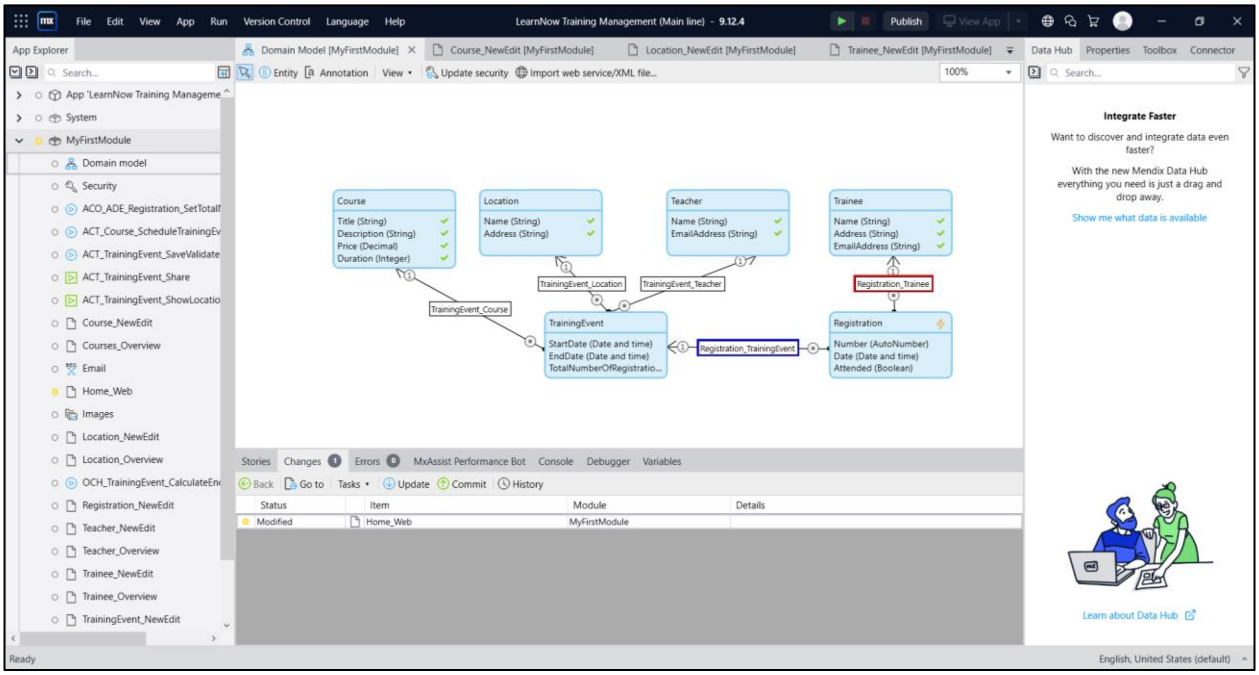
After the creation of the demo application, the main strong suits of SAP Build according to empirical research are the process automation and the end-to-end interoperability within its own ecosystem. The process management in particular is intuitive to use and understand as it follows a step-by-step scheme. The guides and tutorials for process management are well-made and easy to follow. In the SAP Build ecosystem, consisting of SAP Build Apps, SAP Build Process Automation and SAP Build WorkZone, SAP Build Process Automation is the clear focus and the strong suit of the LCNC framework. It also easily connects back into an SAP Build Apps application or into SAP Build WorkZone.

One major weakness in SAP Build for a citizen developer is the learning process. Although there are quite a few guides and tutorials available, as a citizen developer the setup process before the actual LCNC development can take place is cumbersome. THE BTP Cockpit (Business Technology Platform) is difficult to navigate through as a beginner. Furthermore, the errors displayed during the setup phase link to documentation regarding the error, which is good. However, some of this documentation is blocked for users with a trial account. Therefore, some of the documentation regarding SAP Build or the setup cannot be accessed.

Furthermore, the pre-built content for SAP Build Apps and SAP WorkZone is lacking when compared to the SAP Process Automations part of the SAP build LCNC framework. Many fully functional processes can be found in the SAP Build Content Catalog. The SAP Build Content Catalog is a catalogue where pre-built applications and processes can be downloaded and edited to fit the requirements of the user. At the time of writing the catalogue has 388 items, 366 of those are pre-built processes such as a ‘Job Offers Approvals’ business process.

Mendix

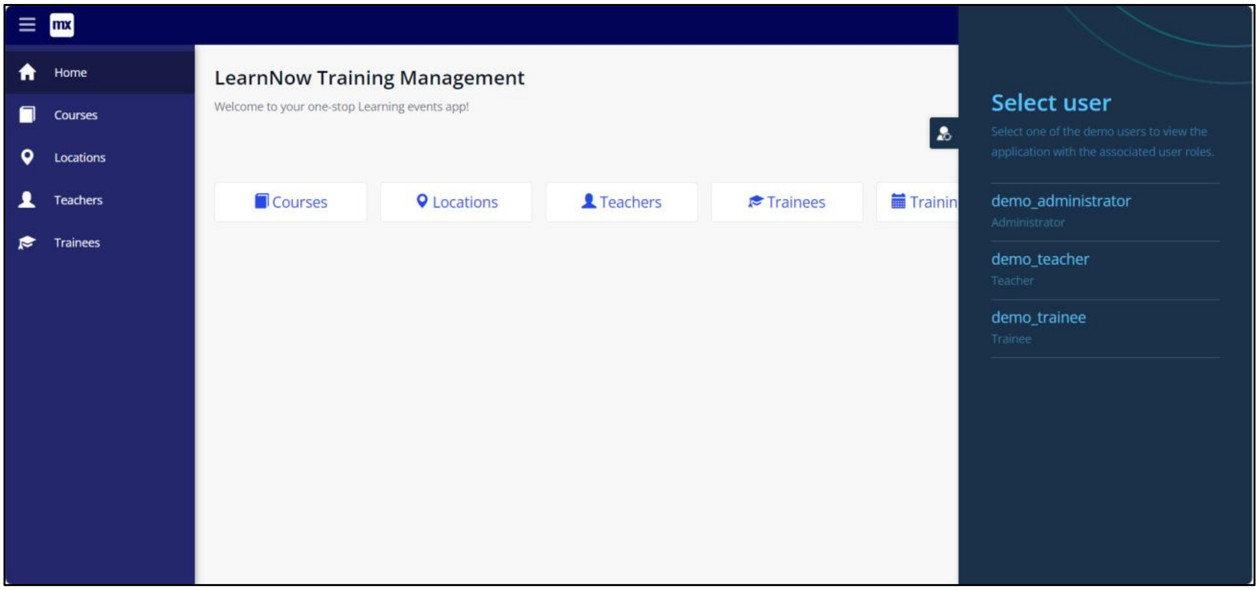
The application built in the LCNC framework Mendix (figure 8) in part follows a comprehensive tutorial of Mendix. Throughout this tutorial, most aspects of creating an application in Mendix are utilised, from Business logic to authentication mechanisms. The demo application built in Mendix Studio is a scheduling and management application for different types of trainings. The demo application can be used to appoint new teachers by the administrator, teacher can add new courses or training locations and schedule said courses and students can use the demo applications to sign up to different trainings. The students can use the applications as a reference for location and timetables after a sign-up. The goal of this application is to test out the overall high standard of functions in the Mendix environment, as well as test out the well-established guides and community support.



Status	Item	Module	Details
Modified	Home_Web	MyFirstModule	

Figure 8 The domain model, or the home page, of the application development in Mendix Studio. Here the functionality of the application is determined. On the left-hand side, the different pages and functionalities made during the development process can be accessed. The console at the bottom is where the changes made can be found, as well as the debugger.

The application can be deployed and tested in a web-browser. The application can be tested from the viewpoint of the different users, as can be seen in figure 9. For certain users, depending on the user role, some parts of the application may not be visible, or the landing page can be entirely different. Therefore, the same demo application can be used by administrators, teachers and students and no separate application needs to be built per user group. For example, a teacher needs to be able to schedule a training session, whereas a student should not be able to.



LearnNow Training Management

Welcome to your one-stop Learning events app!

Select user

Select one of the demo users to view the application with the associated user roles.

demo_administrator
Administrator

demo_teacher
Teacher

demo_trainee
Trainee

Figure 9 The application built within Mendix Studio is tested in a web-browser. On the right-hand side the viewpoint from the three different user roles, administrator, teacher or trainee, can be selected.

Mendix is expansive in both features and guides. Therefore, it is relatively simple to understand if a guide is followed. The design of Mendix itself is less intuitive than other LCNC frameworks in this study at first glance. This can lead people with little digital experience to potentially assume the framework is difficult to learn, however, the consequent Mendix updates during the duration of this study have improve the design of Mendix Studio to look more modern.

A major strength of Mendix is that almost every feature is handled within the same environment, be it the business engine rules, workflow management or GUI development of the app itself. Furthermore, the learning material and the large community of developers of Mendix help citizen developers overcome issues they might encounter.

Furthermore, the Mendix developer portal acts as a portal for the entire project. Here the application itself can be directly opened in the desktop environment, and the collaboration efforts are centralised. The developer portal acts as a team server, where the documentation of the application in question is located, the steps in the process are recorded, and the planning is managed. The team working on the application can clarify changes made to the application and update the progress within the planning itself. This planning, called the ’Story’ can also be updated from within the desktop application Mendix Studio itself. Furthermore, version control of the application in development is also handled in the developer portal.

Logic in Mendix is handled through microflows. A REST API call is handled by the microflow connect to the activity. However, the testing of microflow at the beginning of this study was rather cumbersome, having to launch the app each time, and open a web-browser to test it. During the duration of this study, the Unit Testing Module was released by Mendix, which made the process of testing microflows quicker. However, the Unit Testing Module is not yet easy to use or find for beginners.

Neptune

For Neptune, the empirical research consisted of a demo application made using the Neptune Open edition. The development of this demo application was aimed towards testing the perceived strong suits of Neptune, namely the intuitiveness and the pre-built components. The demo application developed is a dashboard for the New Joiner Experience (NJX) of Accenture, wherein the schedule of the NJX can be seen, links towards helpful sites can be found, as well as how to contact HR or the supervisor. Furthermore, the location of the NJX can be found within the demo application through an integrate google maps functionality.

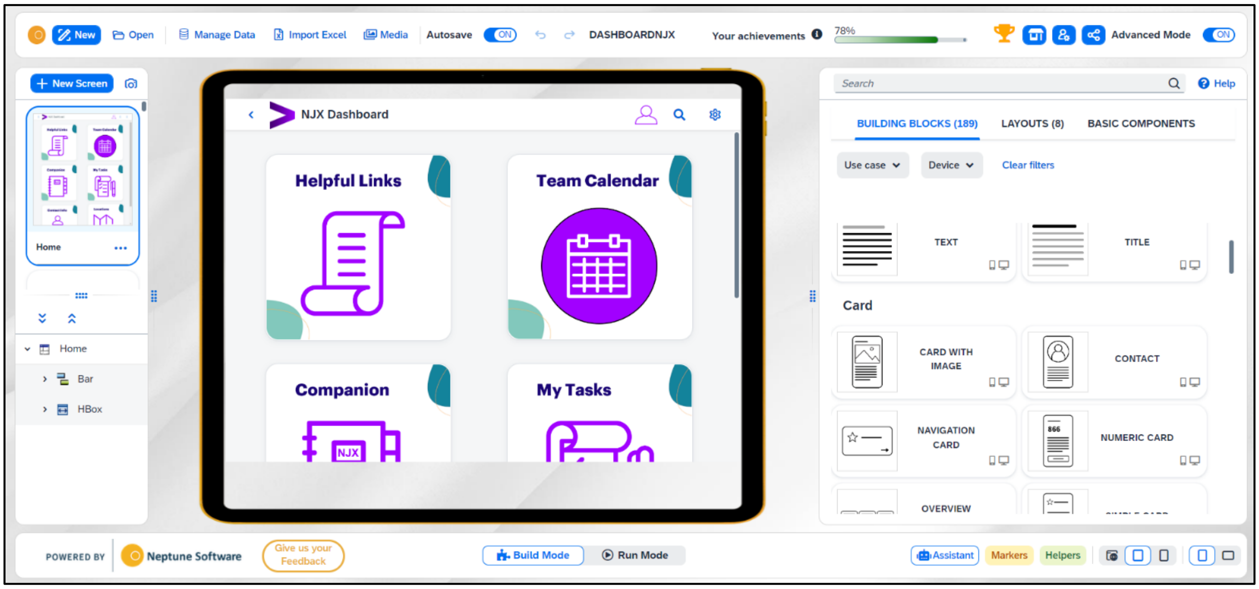
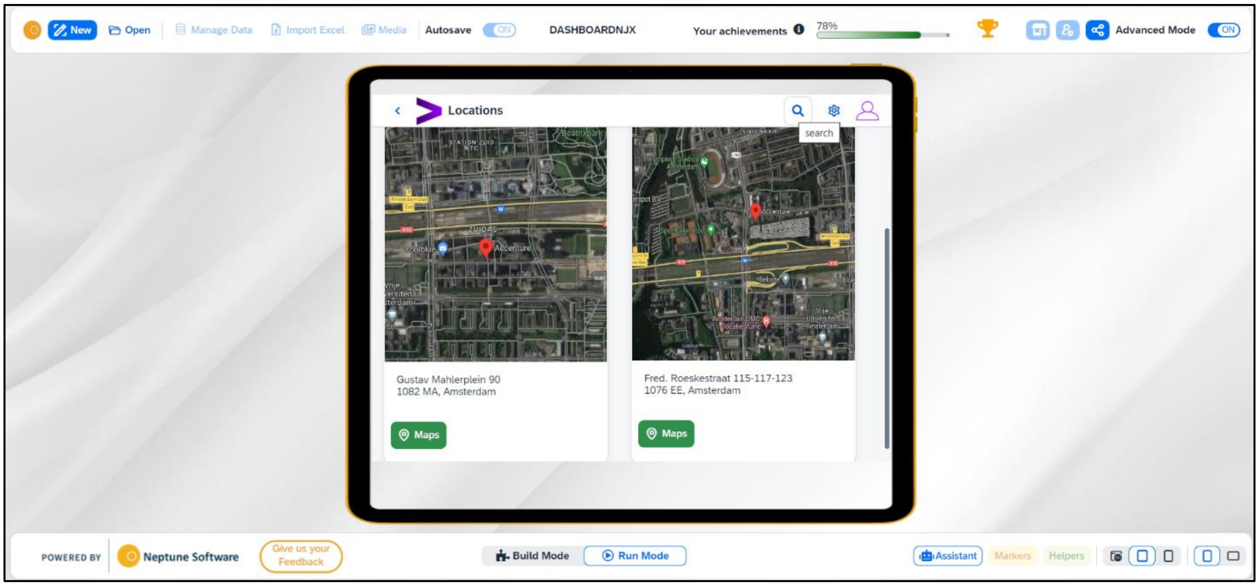


Figure 10 The application built within Mendix Studio is tested in a web-browser. On the right-hand side the viewpoint from the three different user roles, administrator, teacher or trainee, can be selected.

In figure 10 the demo application can be seen in the 'Build Mode'. In this mode the application is built. The pages can be created and selected on the left-hand side. The pre-built content or building blocks can be found on the right-hand side. These building blocks and pre-built content can be added to the application through either drag-and-drop or point-and-click approaches. The application can be very quickly and easily tested by selecting 'Run Mode' in the bottom of the screen. The application can be tested in the same environment, enabling a quick and iterative development process. The 'Run Mode' can be seen in figure 11, where the 'search' button is highlighted and the 'Locations' page of the demo application is shown.



Gustav Mahlerplein 90
1082 MA, Amsterdam

Fred. Roeskestraat 115-117-123
1076 EE, Amsterdam

Figure 11 The NJX dashboard demo application in ‘Run Mode’, the testing environment of the Neptune LCNC platform. In this mode, the application can be quickly tested. All buttons and link which can be seen in this figure, operate in ‘Run Mode’ as if the application is live.

A major perceived strength of the Neptune platform is the intuitiveness and perceived ease of use. This in combination with the pre-built content available can lead to a very rapid development of an application. The NJX dashboard demo application was developed within a couple of hours of work. This could not have been done to the same extent using the other two LCNC frameworks in this study. This ease of use, combined with the sleek and modern design of all elements in the platform, makes it possible for ready-to-use applications to be developed in a relatively short amount of time. This is made possible in part due to the responsiveness and reliability of the platform.

Another strength of the platform is the testing capabilities for the application. The switch between ‘Build Mode’ and ‘Run Mode’ is instantaneous. Furthermore, the addition of ‘heatmaps’ makes bottlenecks easy to find. The application can be live shared with end-users during the development and testing phase of the application to determine how people are using the application. The heatmaps function shows where user spend most of their time on the application. This allows developers to assess what parts of the application users are struggling with most. This is reinforced by users being able to give feedback to specific parts of the application within this same share function.

However, if the to be developed application requires very specific change to be made to pre-built content, the process is more cumbersome. Customisability seems lacking in favour of a modern and intuitive design. Therefore, the ease of use and intuitiveness is a plus, but only if the development occurs within the control limits. Outside of these limits the platform sometimes gets more difficult to use, from the perspective of a new citizen developer at least.

Impact of LCNC on IT world

In this report, three studies / surveys are used to assess the overall impact LCNC platforms have had or are predicted to have on the IT world. The findings of these survey will be cross-reference with available literature on the subject. The surveys used in this study sometimes include no-code and low-code tools under LCNC. This report does not include these tools in LCNC frameworks, therefore the data may be skewed, since the barrier to using these tools in a business is lower than adopting an entire LCNC framework. The surveys and consequently following reports used in this report are the following:

Survey 1

The first survey findings consist of 2 surveys and has been conducted by a third-party medium on behalf of 'OutSystems', a different LCNC company. The studies referenced in this report following the results of the surveys is called 'The State of Application Development – How IT Is Responding to Digital Disruption and Innovation' and 'The State of Application Development – Is IT Ready for Disruption?'. The first survey has over 3300 IT professionals as respondents and was made in the year 2018/2019. The second survey has over 3500 IT professionals as respondent and was made in the year 2019/2020. The surveys and following analysis focus on how app development practices are evolving, what challenges arise when trying to meet application development goals, and how LCNC and citizen development fit into these trends. It also goes more in-depth on the usage numbers of LCNC across sectors and what LCNC is most used for. (OutSystems, 2019, 2020)

The respondents' demographics are also considered in these surveys, as not to introduce a significant bias. The respondents are distributed through career levels and through sectors but are focused within IT (table 5). Furthermore, as was stated in the report, "the survey was promoted primarily to IT professionals who were not OutSystems customers". This further reduces the risk of platform biases being introduced into the survey.

Table 5 The demographics of the respondents of the two OutSystems surveys used in this report. The left table shows the location of the respondents, the right table shows the size of the organisation of the respondents. Adapted from (OutSystems, 2019, 2020).

Geography	Respondents (%)
North America	35%
Europe	34%
Asia and Pacific	17%
South and Latin America	7%
Africa, Middle East, and CIS	7%

Size of Organisation	Respondents (%)
< 500 Employees	50%
500 – 4,999 Employees	25%
5,000 – 10,000 Employees	8%
> 10,000 Employees	16%

Survey 2

The third survey findings consist of the findings of two surveys, as to show the changes in the state of LCNC through time. These surveys have been conducted by DZone, a publisher of knowledge resources for software developers. The first survey is called ‘Low-Code Development – Empowering Business Users, Enabling Developers 2021’ (DZone, 2021). The second survey is called ‘Low Code and No Code – Automation for the Modern Organization 2022’ (DZone, 2022). The first survey recorded 680 respondents globally. The second survey recorded 339 respondents globally.

Survey 3

The third survey findings used in this study consists of the findings of two surveys conducted by Mendix. These two surveys are used to show the changes through time concerning LCNC platform usage. The two surveys in question are called ‘The State of Low-Code 2021: A Look Back, The Light Ahead’ (Mendix, 2021) and ‘Low-Code Verticals Study 2022’ (Mendix, 2022).

The first survey has 2025 respondents, consisting of 1209 IT Leaders and 816 software developers. The second survey had 556 respondents, all Senior Business Leaders in 5 countries: US, UK, The Netherlands, France, and Germany. Both surveys were commissioned by Mendix and governed by an independent research consultancy, thereby decreasing bias (table 6).

Table 6 The demographics of the respondents of the two Mendix surveys used in this report. The left table shows the number of respondents per location for the first survey, the right table shows the number of respondents per locations for the second survey. Adapted from Mendix, (2021) and Mendix, (2022).

Mendix - Survey 1	
Geography	**Respondents**
US	1012
Europe	503
China	510
Total	**2025**

Mendix - Survey 2	
Geography	**Respondents**
US	277
UK	80
Germany	76
France	72
The Netherlands	51
Total	**556**

Usage / Adoption

The first and foremost way in which to measure the impact of LCNC frameworks on businesses is to assess whether the usage of LCNC frameworks has changed over time. The actual usage of LCNC frameworks cannot be easily determined, but the adoption of LCNC frameworks can be seen as 'usage' as a proxy. The overall usage of LCNC under the respondents of the first survey of OutSystems in 2018/2019 ranged from a low of 34% of the businesses using LCNC to a high of 51% of businesses under the respondents using LCNC (OutSystems, 2019). The average usage in the year 2020, found by the is lower than that, with an average of 45% of the respondents having adopted LCNC in some way or form (OutSystems, 2020). In 2021 the usage rose drastically to an average of 79.7% of the respondents, according to an extrapolation both a survey of DZone and Mendix (DZone, 2021; Mendix, 2021). The surveys of Mendix and DZone both noted another increase in LCNC usage in 2022, with the average LCNC usage rising to 89% of the respondents (DZone, 2022; Mendix, 2022). The extrapolation of both the survey results and some supporting literature are shown in figure 12 (Vincent et al., 2019).

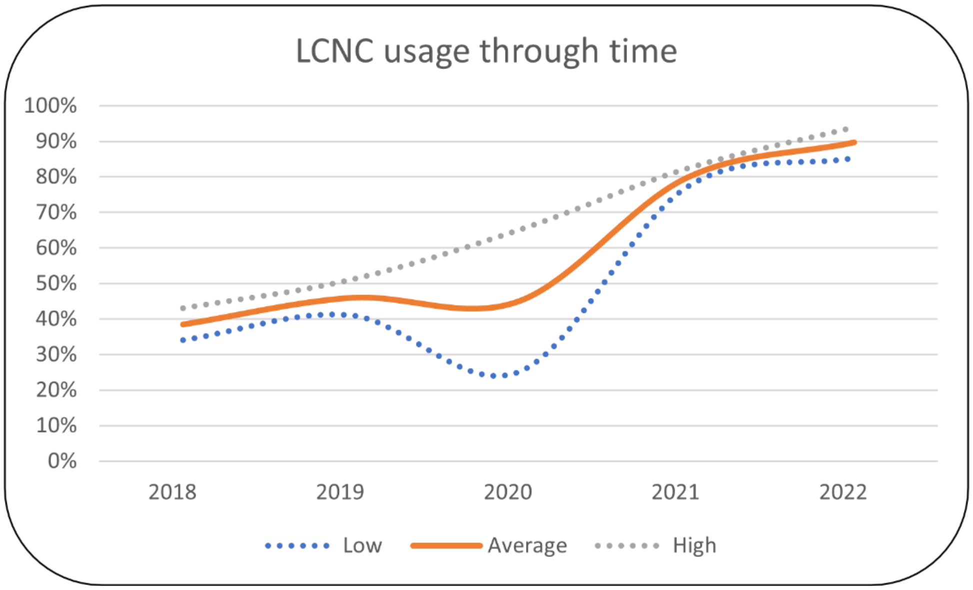


Figure 12 A line chart depicting LCNC usage through time from 2018 until 2022. The combined data of the three surveys of Mendix, OutSystems and DZone is used in this figure in conjunction with available literature on the subject. The 'low' line depicts the only actual usage at the time, whereas the 'high' line includes planned usage for that year.

Main reasons for using LCNC frameworks

The OutSystems survey's findings list multiple reasons as to why respondents started using LCNC. Most of the answers remained the same in both surveys, as can be seen in figure 13. In both surveys the top answers as to why LCNC is being adopted is to 'Accelerate digital innovation and transformation'. However, there was a significant increase in the responses regarding the main reason to adopt LCNC being to increase business responsiveness as well as to reduce the dependency on hard-to-hire software developers. (OutSystems, 2019, 2020)

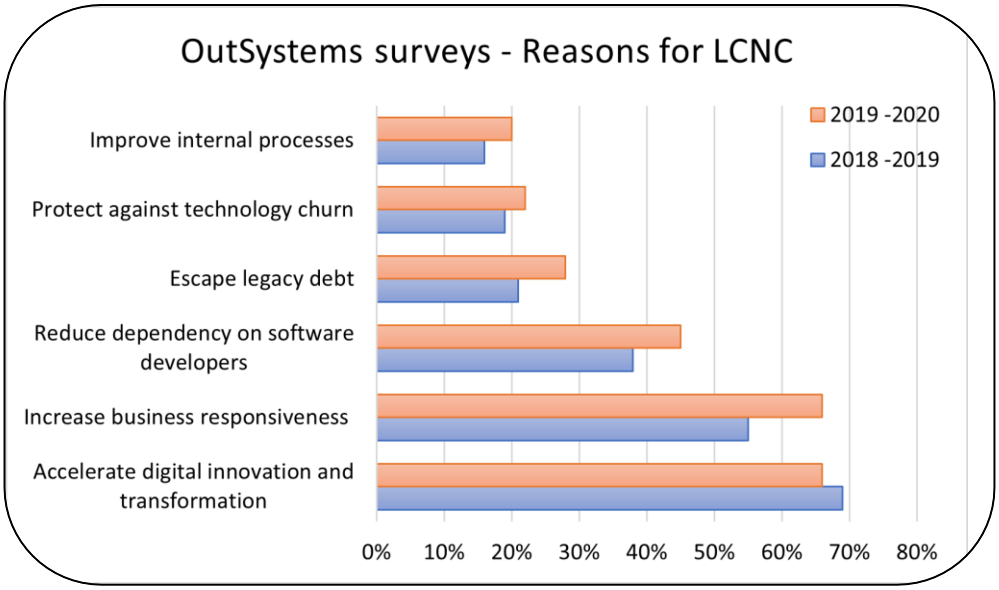


Figure 13 The percentage of respondents' answers when asked what the main reasons for the LCNC adoption is according to the OutSystems surveys in 2018 – 2019 and 2019 – 2020 (OutSystems, 2019, 2020).

The Mendix 2021 and 2022 survey findings for main reasons as to why LCNC is being adopted can be found in table 7. The top reason for implementing LCNC into the business in 2021 was to increase the collaboration between the IT and Business groups with the organisation. In 2022 this was still an important reason, however the top reason changed to be able to quickly respond to business needs in a flexible manner. Faster development and a to relieve the constraint on skill software developers were also important reasons for adopting LCNC frameworks. (Mendix, 2021, 2022)

Table 7 The topmost answered reasons according to the respondents to adopt LCNC as stated in the 2021 and 2022 Mendix surveys (Mendix, 2021, 2022).

Top reasons LCNC - 2021	Top reasons LCNC - 2022
Increased collaboration IT and Business groups	Quickly and flexible respond to business needs
Lower costs of software development	Increased collaboration IT and Business groups
Faster development	Accelerated use of technologies such as AI or IoT
Quickly and flexible respond to business needs	Better management of app-related security
Relieved IT resource constraints	Relieved IT resource constraints

Perceived Drivers and Inhibitors for LCNC adoption

There are multiple drivers and inhibitors for LCNC adoption. The drivers give indication towards why companies choose to adopt LCNC frameworks into their business. The inhibitors can give indication as to why certain companies do not adopt LCNC frameworks, or why these companies are hesitant to do so. A recent study by Käss et al., (2023), assessed drivers and inhibitors of LCNC adoption through semi-structured expert and consultant interviews and subsequent data-analysis methods. The top 5 drivers and inhibitors found in this study can be seen in Table 8.

Table 8 The top 5 adoption drivers and inhibitors for LCNC frameworks according to Käss et al., (2023). The top 5 drivers and inhibitors are gathered from a mean rank of topmost answered drivers and inhibitors between line managers and consultants (Käss et al., 2023).

Top 5 Adoption drivers	Top 5 Adoption inhibitors
Improved efficiency of software development	Lack of LCNC culture and reluctance to change
Reduced entry barriers for software development	Fear of security, compliance and privacy risk
Reduced knowledge required for app development	Lack of governance
Improved business process efficiency	Lack of LCNC developers
Good integration with existing systems	Fear of island application landscape

Another important perceived barrier to LCNC adoption according to a study by Alamin et al., (2023), which researched the most frequent discussion topics regarding the barriers in adopting LCNC frameworks, is related to customisability. According to this study, 30% of the discussions related to barriers in adopting LCNC frameworks are on the customisability or lack therefore in LCNC frameworks. These findings are further reinforced by empirical research done by Abdullah Al Alamin et al., (2021), who also found that most questions from Low-code practitioners relate to customisability. LCNC developers frequently run into issues regarding customisation of user interfaces. According to the interview with a new citizen developer (*Appendix – Interview B*), customisability was also seen as a perceived limitation of LCNC. Furthermore, according to the DZone survey findings, many citizen developers and professional developers struggle with debugging of LCNC-made applications, since the ‘normal’ step-by-step progress of deactivating parts of the code cannot apply.

Development speed

Another measure of impact of LCNC on the IT world can be determined by the effect adoption of LCNC has on the development speed for businesses. According to the survey results of the second OutSystems survey in 2020, the development speed of both web applications and mobile applications increased after adoption of LCNC. The extrapolation of the results found in this study can be seen in figure 14. The development speed of web and mobile applications increased by approximately 13% with LCNC adoption according to the respondents of the OutSystems surveys. The findings of this survey are further reinforced by the findings of the Mendix 2021 survey, stating that the development time of software solutions has increased by 56%, in part due to LCNC adoption (Mendix, 2021).

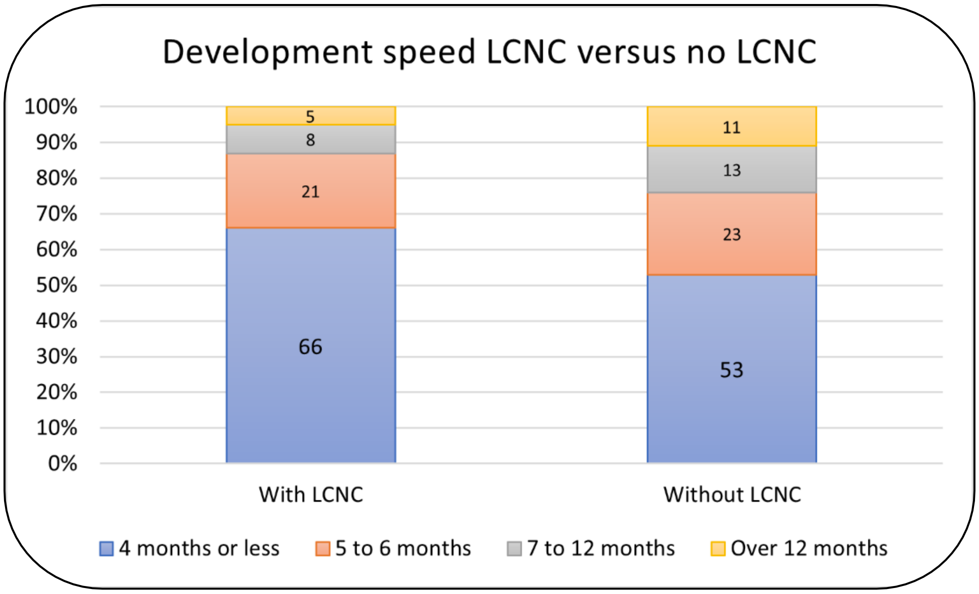


Figure 14 The Web application and Mobile application development speed compared between using LCNC and not using LCNC. The data of this figure is an extrapolation of the findings of the two OutSystems survey assessed in the report (OutSystems, 2019, 2020).

What type of projects LCNC is used for

According to the first OutSystems survey in 2018 - 2019, the main projects LCNC is being used for was employee -and customer-facing portals and web applications (approximately 50% of the respondents reported this use case). This was still the case in the 2019 - 2020 OutSystems survey, although there was an increase in using LCNC to replace or extend existing systems (2019: 31% versus 2020: 38%). These findings resonate with the findings of the 2022 Mendix survey, where the top use cases were found to be Customer portals, productivity apps and extending legacy systems (Mendix, 2022).

According to the DZone surveys, the top 4 most commonly answered use cases found in 2021 and 2022 remaining relatively the same. In the 2021 the top 2 most answered use case were 'Interactive web forms' and 'simple databases', which remained the topmost answered use cases in the 2022. These findings somewhat relate to the aforementioned findings on the use cases of the OutSystems and Mendix surveys. However, according to DZone, 'Business process management' is rising as a most used use case for LCNC frameworks. The top 4 findings of the DZone surveys can be found in table 9.

Table 9 The top 4 most answered use cases in the 2021 and 2022 DZone surveys according to the respondents (DZone, 2021, 2022).

Use cases - 2021	Use cases - 2022
Interactive web forms	Interactive web forms
Simple databases	Simple databases
Request handling	Business process management
Business process management	Request handling

Specific sectors

The LCNC adoption rate in certain markets, sectors and industries is higher than in other sectors, according to the Mendix 2021 and 2022 surveys. Especially the Finance and Banking sector has a high adoption rate. According to the same studies, the public sector has seen a particularly high increase in LCNC adoption between 2021 and 2022 (47% to 95% respectively). The answers to the survey question “Is low-code being used anywhere in your organisation for software development?” can be seen in figure 15. (Mendix, 2021, 2022)

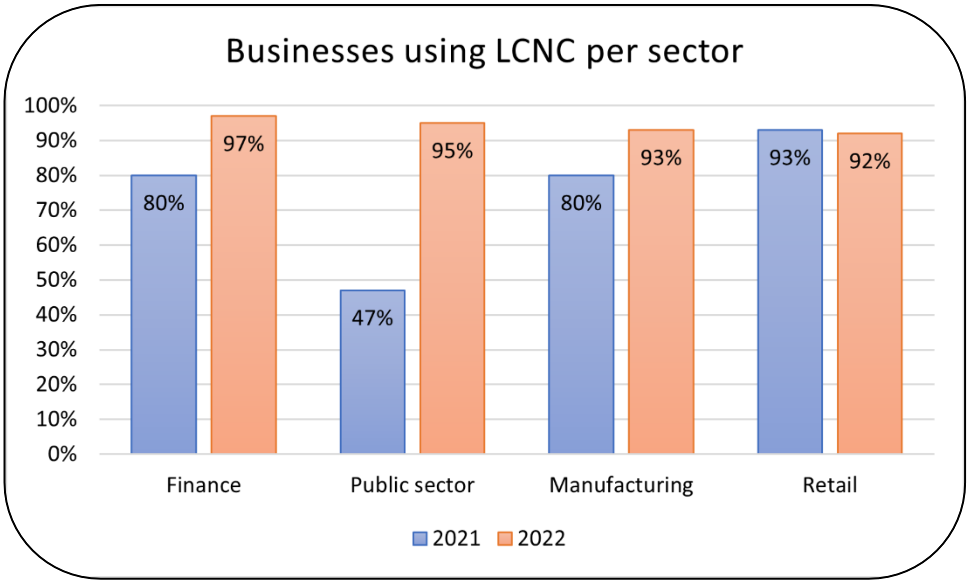


Figure 15 The LCNC adoption per sector given in the percentage of the total respondents. The results to the question “Is low-code being used anywhere in your organization for software development?” in the Mendix survey of 2021 and 2022 is shown in this figure (Mendix, 2021, 2022).

Regional differences

The adoption rate of LCNC differs depending on the region of the world. According to the Mendix 2021 survey, the adoption rate in the US is the highest, with 80% of the respondents of the survey using LCNC frameworks. China LCNC adoption is next, with 75% and Europe has the lowest adoption rate of the respondents with 74%. Another survey held in 2020 by Rackspace reinforces these findings, reporting the US with highest adoption rate of 77% and Europe with the lowest adoption rate with 67% (DeVerter, 2020). More specific data or data on the adoption of the other regions in the world is lacking. The regional differences in adoption of LCNC frameworks can be seen in figure 16.

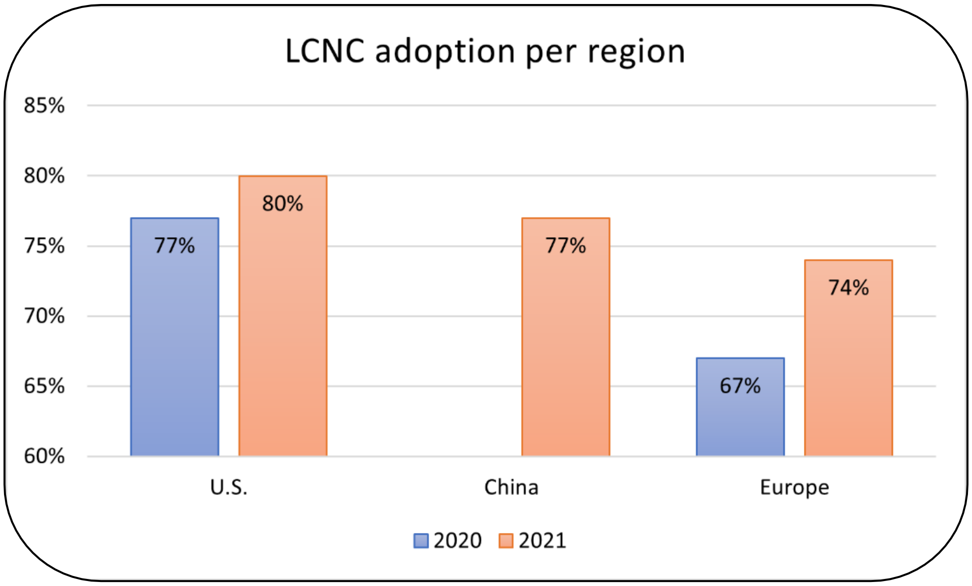


Figure 16 The LCNC adoption per region given as a percentage of the total respondents of the survey. The Mendix survey of 2021 was used for the data of 2021 and a survey by Rackspace was used for the data of 2020(DeVerter, 2020; Mendix, 2021).

Covid-19

The covid-19 pandemic can help give insight into what drives technology adoption. The 2021 Mendix survey states that 64% of the organisations in this survey relied on non-IT employees to help the IT departments during the increase in workload due to the pandemic (Mendix, 2021). In accordance with this, approximately 83% of the respondents' state that their organisations started using LCNC since the covid-19 pandemic, where 38% of the organisation adopting LCNC used it for mission-critical solutions (Mendix, 2021). In the 2022 Mendix survey, 69% of the respondents' state that LCNC has evolved from a crisis solution during the pandemic, to a core technology now (Mendix, 2022). A further 10% of these organisations see LCNC as the foundation of their business, according to the same survey (Mendix, 2022).

Discussion / Conclusion

How do the three LCNC frameworks, SAP Build, Mendix and Neptune, compare?

The comparison between SAP Build, Mendix and Neptune based on their features has multiple answers. Based on the comparison method alone, Mendix has the highest potential with a total score of 82 out of 92. SAP build and Neptune score significantly lower with a respective score of 63 and 62. These results show that Mendix scores the highest average on all features. According to this study, Mendix therefore is the strongest LCNC framework when it comes down to the number of features and the quality of the implementation of these features. However, there are certain features which each of the three LCNC framework excels at, as gathered from the *results – Strengths and Weaknesses*. SAP Build excels at its interoperability. SAP Build can integrate many different types of databases and data structures. Furthermore, it there is native integration into other SAP systems, which has the added benefit of increased security (*Appendix – Interview A*). Neptune excels at reusability of its components and the quality of the pre-built components. Mendix, as stated before, has many strengths and therefore excels at being adaptable and useful to many different development scenarios.

It must be noted that the scoring method of this report does not necessarily distinguish between the value or usefulness of specific features. Therefore, a LCNC framework with a lower score, does not necessarily mean that the specific LCNC is worse than the other. It can also mean that the specific feature is scored as being 'absent' (score of 0), even though it does not necessarily indicate that the feature is not available in one form or another at the LCNC framework. It could be the case that it was not readily available at the time of writing this report, that the feature was not easily accessible, referenced to or that literature or guides could not be found on that specific feature. If a LCNC framework scores mediocre at most features, but excels at e.g., process management, it can still be more effective at certain tasks than higher scoring LCNC frameworks. This same can be said for a feature such as cost of a LCNC framework, which falls under the feature group 'Accessibility'. The weight of cost is the same as every other sub-feature, but it can play a large role in whether a business or client adopts the framework. This, however, depends entirely on the budget the business or client has allotted to adopting a LCNC framework, which varies depending on a lot of variables, such as size of the company. Therefore, a different weight is difficult to assign to the sub-feature in this method since the research is not poised towards a specifically sized business.

However, some features such as cost, can be determined to be more important than others for a business when choosing a LCNC framework. Some of these important features will be discussed in this part of the report. One of those features is security, because if a LCNC framework is not secure, all the other features and functions it has do not matter. However, all of the frameworks in this study are relatively safe to use and secure, owing to the security-by-design philosophy that most LCNC frameworks adhere to. This means that every step of development in one of these LCNC frameworks takes security into account. However, security can be more robust when all connections and destinations are restricted to the same environment, as can be achieved with SAP Build, by staying with the SAP environment (*see Appendix – Interview A).* SAP Build has more than optimised security owning to the native integration within SAP systems themselves, since all the developments stay within the same environment (Interview A – Appendix). However, this is only the case if the LCNC framework is integrated into an SAP environment.

Another of those important features can be collaborative development support, since it can be a big draw to LCNC for businesses. According to the *Results - Main reasons for using LCNC frameworks*, increasing collaboration between IT and Business groups is seen as a big reason to adopt LCNC. SAP Build is lacking in this feature group, with only a quarter of the maximum points scored in the comparison method. This is a weakness of SAP Build, because if businesses are seeking to increase the collaboration between IT and Business groups, this can be a feature that cannot be overlooked, even if the rest of the features are a fit.

AI assisted development is a feature which can potentially be seen as essential in this rapidly change business landscape. AI assisted development is available in both Neptune and Mendix but lacking in SAP Build. SAP Build is most likely less quick to adapt these new developments due to the size of the entire environment. SAP Build needs to be able to tap into the entirety of the SAP environment and therefore needs some form of (backwards) compatibility. Furthermore, the decision to include AI into the process is made higher up the chain of command, since SAP Build is not a standalone product on itself, whereas Mendix and Neptune are both LCNC frameworks entirely (*Appendix – Interview A*).

Lastly, an important 'feature' of a LCNC framework can be how it 'feels' to use. This is in part related to the ease of use and intuitiveness of the framework. However, it can be more personal than those features. The 'feel' of a LCNC framework can be different for every citizen developer, depending on what types of programmes he or she uses regularly or the background of the citizen developer. The 'feel' of a LCNC framework cannot be determined beforehand. Therefore, it can be important to compare the different LCNC frameworks using the free or trial version before a decision is made solely on the quantifiable parameters of each framework. All three LCNC frameworks in this study have a trial or free version of their platform available.

When comparing the three different LCNC frameworks from the perspective of a clean, new enterprise software environment, the comparison method used in this report can be used to aid in decision making. However, if a specific software environment is already in place, decision can be skewed in certain directions depending on compatibility. From the perspective of an SAP enterprise software environment, SAP Build is difficult to replace. Mendix and Neptune both have their pros and cons, however, the intrinsic security of using native integration with SAP Build into the other SAP systems is hard to beat. Furthermore, through the use of destinations, more or less everything with the SAP ecosystem is possible. According to *Appendix – Interview C,* the SAP edition of Neptune can also be used to integrate into SAP systems natively, as has been done by a collaboration between Accenture and BlueWorx. However, during the writing of this report that has not been tested.

All in all, the comparison between the three LCNC frameworks boil down to the different preferences or needs of a business. For quick, simple and sleek app design, Neptune is better. For overall reliability and functionality Mendix is better, but for using a LCNC framework within an already established SAP system, SAP Build is hard to beat.

One thing to consider that SAP Build consists of three separate LCNC platforms, SAP Build Apps, SAP Build Process Automation and SAP Build WorkZone. Because of this it is difficult to accurately compare the frameworks. Furthermore, the qualitative comparison method scored the features present for entire SAP Build environment. The three parts of SAP Build would have scored much lower separately since they work in conjunction with one another. This is made difficult however, because the GUI features of SAP Build Apps for example, might score differently than SAP Build Process automation. With all three parts of SAP Build together, the end-to-end capabilities of the framework are higher than each standalone part.

How big / disruptive is the impact of LCNC frameworks in the IT world?

The impact of LCNC frameworks on the IT world is difficult to accurately measure. The first point of contention comes from the fact that 'impact' can consist of a variety of parameters. The first and most general parameter that can be looked at to quantify the impact of LCNC frameworks is the actual usage of LCNC frameworks by organisations through time. As can be seen from the *Results - Usage / Adoption*, the usage of LCNC technologies has increased over time, from an average usage of 39% of the questioned organisations in 2018, to an average usage of 89% in 2022. These findings suggest that Low-code No-code is acquiring a significant foothold in the IT world and has already bridged the chasm in the adoption curve.

However, the usage through time graph is by definition not entirely accurate. The difference in definitions between the different surveys and literature can lead to a misrepresentation of the actual data. Particularly the different definitions of the term LCNC between the surveys can explain the dip in the usage of LCNC in 2020 according to the data shown in figure 12. Some surveys measure LCNC usage including LCNC tools, whereas other surveys only consider the usage of actual LCNC platforms or frameworks. This difference in definition can have a large impact on the data, seeing as LCNC tools are much easier to adopt than an entire LCNC framework. Furthermore, since the findings of the results rely on multiple surveys, conducted by different organisations which not necessarily ask the same questions, the results again, can be somewhat unreliable. However, seeing as Gartner in the Magic Quadrant predicted that over 50% of medium to large businesses will have potentially adopted LCNC frameworks by 2023, and the more stringent definition usage in the survey in 2020 was already close to that adoption, the assumption can be made that the LCNC usage is at least rising in usage numbers and having an impact on the business world (Vincent et al., 2019).

Another manner in which the impact of LCNC frameworks in the IT world can be assessed is by looking at whether LCNC is used for any business-critical development, or even whether the survival of an organisation depended upon LCNC frameworks. The covid-19 pandemic can give a good indication towards this. According to the *results – Covid - 19*, 83% of the organisations questioned during a survey stated that they increased the LCNC usage during the pandemic. In the same survey a year later, 69% of the respondents stated that the LCNC framework usage within their organisation evolved from a crisis solution to a core technology. Furthermore, a further 38% of these organisations use LCNC for mission-critical operations, out of which 10% state that LCNC has become the foundation of their business. This leads to believe that the pandemic has been a catalyst for many organisations to adopt LCNC frameworks into their business strategy, thereby driving LCNC up the adoption curve. The further retention of the LCNC frameworks suggests that once LCNC has been embraced into the digital structure of an organisation, it has the potential to be a become a core part of the IT structure of said organisation. Therefore, according to this manner of measuring impact, LCNC frameworks have a significant effect on the IT world.

A third manner in which the impact of LCNC on the IT world can be determined, is by assessing for what purpose the LCNC platforms are being used. According to the *results - Main reasons for using LCNC frameworks*, multiple survey findings found that one of the main purposes the LCNC is being used for is accelerating digital innovation and the digital transformation as well as to increase the responsiveness of the business. This suggests that a large number of organisations which have adopted LCNC into their business strategy, utilise it to remain competitive and keep control over the direction of their business. Which in this fast-moving digital business landscape is key to survival, owing to the term 'Innovate or Die'.

Combining all the different definitions and manners in which impact can be measured, it is clear that LCNC frameworks are being used for business-critical development and innovation. Furthermore, the adoption rate of LCNC frameworks is increasing. These results together suggest that LCNC frameworks are having a large impact on the IT world. It is clear however, that the research on impact and adoption of LCNC frameworks is lacking. Most papers and literature available on LCNC frameworks have been provided by practitioners, with academic research lagging behind. Uniformity within research in definition will increase as academic research on the subject increases, thereby increasing the ability to measure the impact of LCNC on the IT world.

How can the strong suits of each framework be leveraged to maximise impact?

The strengths of each framework can be leveraged to maximise impact through assessing which types of projects and by which sectors LCNC is most used for. These types of projects can then be cross-referenced with the strengths of each LCNC framework in order to maximise the impact of the LCNC framework on that project or sector. For example, according to the *results - What type of projects LCNC is used for*, a lot of use cases for LCNC currently are employee-facing portals and customer-facing portals. Depending on the desired implementation of these portals, different LCNC frameworks might suit the needs best. If it is preferred for both the employee-facing portal and the customer-facing portal to be the same application, then the strengths of Mendix might suit this use case. Utilising Mendix, a single application can easily be made where, depending on the assigned user role, the application works differently for employees than it does for customers. However, if a modern and intuitive design is required for both use cases, Neptune can provide these functionalities. Alternatively, according to the same results section, an increase in the desired use case of business process management between 2021 and 2022 can be seen. This use case fits SAP Build better since process management is a strength of SAP Build.

As mentioned above, assessing the adoption rate of different sectors or markets in conjunction with the types of projects most used by those sectors, can also help to determine how the strengths of a LCNC framework can be leveraged to maximise impact. According to the *results – Specific* sectors, the public sector is currently seeing a rapid increase in LCNC adoption. Since the public sector handles a large number of requests and operates a large amount of public web forms, the strengths of a LCNC framework which are compatible with those needs can be utilised. Mendix is the most well-rounded LCNC framework out of the three in this study, so according to the multiple needs of the public sector, Mendix can potentially be leveraged the easiest to maximise the impact in that sector. However, if more importance is given to the automation of the processes behind these forms, such as request handling or application approval / denial, the strengths of SAP Build will be better suited.

Therefore, to maximise the impact of each LCNC framework, multiple variables need to be considered. The most important types of projects of a business, but also to what sector it belongs to enable interoperability between organisations. Furthermore, the impact of a LCNC framework depends on the legacy system in place in the organisation in question. If the IT system already in place is SAP, the safest option to guarantee impact will potentially be SAP Build. However, this is still dependent on the main need of the organisation. If an organisation has an SAP environment in place, but the main draw and need of a LCNC framework for that organisation is web forms, then SAP Build might not be the best course. Neptune could maximise the impact via the strength in reusability, or Mendix could be the better fit due to the strengths in maintenance support and accessibility.

Depending on the sector and needs of a business, a very important aspect to maximise the impact of a LCNC framework can be to assess the future potential needs, opposed to the current needs. For example, according to Bratincevic et al. (2023), generative AI is both an immense opportunity and a large uncertainty for the LCNC market. If the business in question is part of sector wherein generative AI could potentially play a large role in the future, the decision to leverage an LCNC to maximise the impact now, could potentially hurt the business if the strength of the chosen LCNC framework does not include AI. Mendix and Neptune both have already included AI functionality within their LCNC frameworks. However, SAP Build is still lacking in the AI department. According to SAP (*Appendix - Interview A*), generative AI is being looked at within SAP, and SAP Build has plans of integrating AI driven development into the LCNC framework. But these kinds of decisions are made higher up in the SAP chain of command. Whereas other, smaller, or more singularly-focused LCNC frameworks, such as Neptune or even Mendix, have the agility to implement AI features much more quickly. Whereas the company SAP has much more than only LCNC to account for, Mendix and Neptune can focus a larger portion of their attention into integrating AI capabilities into their platforms, with lower risk to already implemented systems of businesses (SAP environments). Therefore, leveraging the strengths of a LCNC framework now irrespective of the future needs of the company, can have a positive impact on the current health of a business, but a negative impact on the futureproofing of a business. This demonstrates the importance of correctly assessing the needs of a business, current and future, in accordance with the strengths of a LCNC.

In what aspects can Accenture clients benefit from using LCNC frameworks?

Accenture clients can potentially benefit from using LCNC frameworks. The exact benefit depends largely on the needs of the client, the sector the client is active in and the current business landscape. If the needs of the client have overlap with strengths and use cases of a specific LCNC framework, the potential benefits can be rather large. According to the *results - Development speed* and the surveys, the benefits of a correctly implemented LCNC framework within the business strategy can increase the development speed of applications up to 56%, reduce the general costs of the applications by 53% and by extension increase the revenue of the application in question.

The use cases found in the *results - Main reasons for using LCNC frameworks*, can give insights into what types of use cases businesses generally consider LCNC frameworks to be beneficial for. For example, if a client of Accenture is in dire need of efficiency and needs to be able to adapt to the ever-changing business landscape more quickly, then adopting a LCNC framework into the business strategy can provide a considerable benefit. Whether adopting a LCNC framework is the right choice, depends on the specific challenges, priorities, budget and future needs that a client has.

Furthermore, the specific benefit a LCNC framework can have for an Accenture client can also be highly dependent on the sector of said client. In the financial sector for example, 63% of the Mendix survey respondents belonging to this sector consider LCNC to have a high potential of simplifying complex financial systems, platforms and processes. In the Manufacturing, Production, or Industry sectors, the largest (perceived) benefit of LCNC is in modernizing legacy IT and switching to mobile applications, as well as production monitoring and execution. If an Accenture client in this sector is facing the same challenges, then a LCNC framework can potentially provide these benefits. In the Government or Public sector for example, the challenges that LCNC is being used to overcome are Improving security, coping with the volume of data collection, and bettering integrating data & process. These are some of the aspects in which Accenture clients can benefit from adopting a LCNC framework.

Why and how should Accenture position LCNC frameworks to its clients?

Accenture should position LCNC frameworks as a potentially untapped resource to its clients, with the potential of belong to the early majority on the adoption curve. One of the reasons for this is the potential benefit clients can have due to the regional differences in LCNC adoption. According to the *results - Regional differences*, Europe lags behind when it comes to digitalisation and LCNC adoption. Accenture could potentially position LCNC frameworks to her clients as a chance to gain a competitive advantage over the direct European market competition by adopting it now. By adopting LCNC sooner, a potential head start on the direct competitors of the business in question can be gained on the path of digital disruption. This is further reinforced by the survey results stating that 72% of the IT Leaders globally state that LCNC is a trend that cannot be afforded to miss. Considering the cost of LCNC frameworks and the potential loss of missing the digital disruption curve, the benefit-cost ratio of investing into a LCNC framework now is potentially large.

Furthermore, if Accenture invests in acquiring in-house knowledge of different LCNC frameworks, clients could be guided by Accenture during the adoption phase of a LCNC framework. If more clients of Accenture adopt LCNC frameworks, this could potentially free up more Accenture developers for large projects such as digitalisation or system migration for other clients. Furthermore, another benefit of acquiring in-house knowledge on LCNC is to be able to take away the general inhibitors to LCNC adoption and show the actual benefits it can have. According to *results - Perceived Drivers and Inhibitors for LCNC adoption*, Lack of LCNC culture and a reluctance to change is the topmost inhibitor to LCNC adoption. This inhibitor can be overcome by being able to show the benefits of a LCNC framework from first-hand experience and being able to assist in the adoption process. Through in-house knowledge of LCNC most of the inhibitors as mentioned in the results can be taken away since most inhibitors are based on fear and lack of knowledge. Accenture can also provide guidance and courses to enable clients to turn employees into citizen developers, opening clients up to the potential benefits such as cost savings and an increase in development speed.

However, Accenture should be careful not to position LCNC frameworks as a one-size-fits-all solution. LCNC frameworks can be beneficial, but there is still a need for people who understand the mechanics of how the LCNC framework functions, as well as people with the insight to know when and where LCNC is beneficial to use. LCNC platforms cannot be used for every IT related challenge that a company might face. Furthermore, Accenture should make it clear to the client that there are certain drawbacks to using LCNC frameworks, such as the lack of customisability. According to the *results - Perceived Drivers and Inhibitors for LCNC adoption,* most questions from LCNC practitioners relate to customisability. LCNC developers frequently run into issues regarding customisation of user interfaces. This demonstrates a potential problem clients can run into when using LCNC frameworks. Because LCNC frameworks are to be used in a simplified manner, e.g., through visual, drag-and-drop functionalities, customisability is lost. Furthermore, this extends to debugging as well, where many citizen developers and professional developers struggle with debugging of LCNC-made applications, since the 'normal' step-by-step progress of deactivating parts of the code cannot apply. According to the interview with a new citizen developer (*Appendix – Interview B*), customisability was also seen as a limitation of LCNC. However, after becoming more comfortable with the specific LCNC framework, this was no longer the case. The lack of customisability was still sometimes perceived as annoying, but mostly it also served as a way to uniformly design applications and reduce errors, since everyone who works on the application knows and adheres to the same format automatically.

All in all, Accenture should position LCNC frameworks to clients as a solution to software development related challenges, with a potentially large benefit-cost ratio. Accenture should acquire the knowledge to assist its clients with the LCNC adoption process and help with the educating citizen developers. However, Accenture should also be clear to its clients on the drawbacks of LCNC frameworks, such as a lack of customisability.

Reflection

My aim with the business internship was to gain more knowledge on how the business world operates. Furthermore, since I used Python extensively during my Major Research Project and enjoyed it, I wanted to explore and learn more about the IT world, especially in the context of business. This internship was therefore a great fit to my aims, especially since I have no IT background and the general appeal of LCNC is that little IT knowledge is required to start using it. The only checkbox that was not ticked during the internship was a connection to sustainability.

The internship in itself was quite a challenge, seeing as I had no previously experience with the 'real' business world or IT. Adjusting to an entirely new environment, learning three different LCNC frameworks and learning a lot about the IT world; these are some of the things I had to do during this internship. The adjustment from a purely academic environment (university) to a business environment (Accenture) was quite large. Furthermore, the adjustment from Life sciences to Business IT was even larger. During this internship I have had to learn (as a citizen developer), three different LCNC frameworks from the ground up, learning about the IT world in general, learn about the business world in general and assess the impact LCNC has had on this newly acquired information on the business world. It has been quite a lot to take in, especially in 6 months of time. However, I did enjoy the process of learning something completely different.

The internship itself did not have many specific connections to the FBE courses. The overlap with the courses was much more on a higher level than on a specific part of a course. The overlap mainly pertains to the mindset and the perspective that I learned during the courses. For example, how to look at different tools and innovations and how these could be leveraged to gain a competitive advantage. These teachings from the courses helped me to draw conclusions or to assess the worth of features and tools.

For the duration of this internship, I believe I performed adequately. Some things could have gone better but considering the experiences in my personal life I had during this internship, which made me lose quite a lot of time within the normal working hours and added onto stress, I think I performed well. I learned an entirely new aspect of the business world, namely IT, and more specifically LCNC development. This knowledge was more difficult for me to learn because I had no previous IT experience, and because it is a rather novel approach to application development, so there weren't many people who could teach me. I learned it well enough to teach a course with two colleagues to other Accenture employees, and semi-hosted an SAP Build course by Daniel Wroblewski (SAP) called Code Jam. Therefore, I can only assume that the knowledge I attained was at an adequate level.

All in all, I think the internship was a decent fit, seeing as I have always had an interest in the digital world, along with an interest in the Business world and consultancy. The internship did not have any link to my scientific interests, such as sustainability or marine sciences, but I took a scientific approach to the empirical research in this internship. This made it still feel more connected to science than it potentially would have been otherwise. It has been a very educational experience for me because I got to experience the 'real' business world, in a setting which interests me. I also learned a lot about having a professional mindset and expanded my network with a lot of interesting and knowledgeable people. The process of this internship leads me to believe that I made the right decision when choosing the master Science and Business Management, and has helped me narrow down the scope of what I want to do for a living.

Demo application reflection

I could not test certain aspects of LCNC frameworks as well as I would have liked. Because there was no access to backend systems or SAP environments to connect to / integrate and test, some parts of the LCNC frameworks were tested in limits. For example, the integration of Neptune within an SAP environment was very difficult to test, since no SAP environment to integrate into was available. Furthermore, the overall testing capabilities in this study were limited to a 'new' citizen developers' perspective. There was little time to familiarise myself with all the frameworks and adopt a developers' mindset. This can also be seen as a plus, since the findings of this study do reflect a freshly 'formed' citizen developer, however, because the main focus of this study was to analyse and assess, the progress in the area of citizen development did not follow its natural course. If the duration of this study was instead primarily focused on learning how to utilise LCNC frameworks for the sake of using them, the findings of this report might have been different. This is partly due to the fact that some LCNC frameworks need more time to understand and familiarise with than others.

However, this also demonstrates how this internship was Inherently biased due to its. Working from an SAP partner, SAP environment. Furthermore, familiarising myself with three different frameworks cannot be done in a uniform manner. Not only because the guides and tutorials are different, but also because the last framework to familiarise myself with will have a benefit because having already familiarised myself with two other LCNC frameworks means I already know the general structure and feel of LCNC frameworks. Also, because these frameworks are all structured differently, in both appearance and function, a comparison based on personal preference will also automatically occur.

However, all in all, the creation of the demo applications was a good way in which to gain experience with all of the three LCNC frameworks in this internship.

Reference list

Abdullah Al Alamin, M., Malakar, S., Uddin, G., Afroz, S., Bin Haider, T., & Iqbal, A. (2021). An Empirical Study of Developer Discussions on Low-Code Software Development Challenges. *Institute of Electrical and Electronics Engineers*. https://doi.org/10.1109/MSR52588.2021.00018

Alamin, M. A. Al, Uddin, G., Malakar, S., Afroz, S., Haider, T., & Iqbal, A. (2023). Developer discussion topics on the adoption and barriers of low code software development platforms. *Empirical Software Engineering*, *28*(1), 4. https://doi.org/10.1007/s10664-022-10244-0

Melnichuk, A. (2022). *Current state of the software developer job market*. NCube.

Beręsewicz, M., Cherniaiev, H., Mantaj, A., & Pater, R. (2023). Text analysis of job offers for mismatch of educational characteristics to labour market demands. *Quality & Quantity*. https://doi.org/10.1007/s11135-023-01707-7

Bernsteiner, R., Schlögl, S., Ploder, C., Dilger, T., & Brecher, F. (2022). *Citizen vs. Professional developers: differences and similarities of skills and training requirements for low code development platforms*. 4257–4264. https://doi.org/10.21125/iceri.2022.1036

Binzer, B., & Winkler, T. J. (2023). Low-Coders, No-Coders, and Citizen Developers in Demand: Examining Knowledge, Skills, and Abilities Through a Job Market Analysis. *Wirtschaftinformatik 2023 Proceedings*, *17*.

Bock, A. C., & Frank, U. (2021). Low-Code Platform. *Business & Information Systems Engineering*, *63*(6), 733–740. https://doi.org/10.1007/s12599-021-00726-8

Bratincevic, J., Gardner, C., Condo, C., Mooter, D., Cornwall, A., Dickerson, D., Morana, S., & Hartig, K. (2023). *The Forrester Wave^{TM}: Low-Code Development Platforms For Professional Developers*. https://reprints2.forrester.com/#/assets/2/225/RES178497/report

Breaux, T., & Moritz, J. (2021). The 2021 software developer shortage is coming. *Communications of the ACM*, *64*(7), 39–41. https://doi.org/10.1145/3440753

Carroll, N., & Maher, M. (2023). How Shell Fueled Digital Transformation by Establishing DIY Software Development. *MIS Quarterly Executive*, *22*(2).

Carroll, N., Ó Móráin, L., Garrett, D., & Jamnadass, A. (2021). The Importance of Citizen Development for Digital Transformation. *Cutter Business Technology Journal*, *34*(3), 5–9.

Danhieux, P. (2022). Navigating The Developer Shortage Crisis: A Time To Define The Developer Of The Future. *Forbes*.

de Visser, H. (2023). *Coming in 2023: The Merging of Studio and Studio Pro*. Mendix. https://www.mendix.com/blog/coming-in-2023-the-merging-of-studio-and-studio-pro/

DeVerter, J. (2020). *The State of Low-Code/No-Code Development Around the Globe*.

DZone. (2021). Low-Code Development - Empowering Business Users, Enabling Developers. *DZone Trend Report*.

DZone. (2022). Low Code and No Code - Automation for the Modern Organization. *DZone Trend Report*.

European Investment Bank. (2023). *Digitalisation in Europe 2022–2023 - Evidence from the EIB Investment Survey*.

Gurcan, F., & Taentzer, G. (2021). Using Microsoft PowerApps, Mendix and OutSystems in Two Development Scenarios: An Experience Report. *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 67–72. https://doi.org/10.1109/MODELS-C53483.2021.00017

Hyun, C. Y. (2019). Design and implementation of a low-code/no-code system. *International Journal of Advanced Smart Convergence*, *8*(4), 188–193.

Käss, S., Strahringer, S., & Westner, M. (2023). Practitioners' Perceptions on the Adoption of Low Code Development Platforms. *IEEE Access*, *11*, 29009–29034. https://doi.org/10.1109/ACCESS.2023.3258539

Khorram, F., Mottu, J.-M., & Sunyé, G. (2020). Challenges & Opportunities in Low-Code Testing. *Association for Computing Machinery*. https://doi.org/10.1145/3417990.3420204

Lavelle, J. (2019). Gartner survey shows global talent shortage is now the top emerging risk facing organizations. *Gartner*.

Lebens, M., & Finnegan, R. (2021). Using a Low Code Development Environment to Teach the Agile Methodology. In C. and W. X. and K. P. Gregory Peggy and Lassenius (Ed.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 191–199). Springer International Publishing.

Lebens, M., J Finnegan, R., C Sorsen, S., & Shah, J. (2021). Rise of the Citizen Developer. *Muma Business Review*, *5*, 101–111. https://doi.org/10.28945/4885

Liu, X., & Zhu, W. (2022). Design and implementation of an online registration system based on Siemens low-code platform. *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, 798–801. https://doi.org/10.1109/AEMCSE55572.2022.00160

Marc, H., & Mileff, P. (2023). Introduction to Mendix. *Production Systems and Information Engineering*, *11*(3), 64–84.

Mendix. (2021). A Look Back, The Light Ahead. *The State of Low-Code*.

Mendix. (2022). Low-Code Verticals Study. *The State of Low-Code*.

Neptune. (n.d.-a). *Neptune DXP - Open Edition*. Neptune Software. Retrieved August 10, 2023, from https://www.neptune-software.com/neptune-dxp/open-edition/

Neptune. (n.d.-b). *Neptune DXP - SAP Edition*. Neptune Software. Retrieved August 10, 2023, from https://www.neptune-software.com/neptune-dxp/sap-edition/

OutSystems. (2019). How IT Is Responding to Digital Disruption and Innovation. *The Sate of Application Development*.

OutSystems. (2020). Is IT Ready for Disruption? *The State of Application Development*.

Ozkaya, I. (2022). The Developer Nation. *IEEE Software*, *39*(1), 3–6. https://doi.org/10.1109/MS.2021.3118481

Pham, T. (2021). Analyzing the software engineer shortage. *Forbes*.

Raimi, L. (2021). Human Capital Development through Reinventing, Retooling and Reskilling Strategies. *Conference Towards ASEAN Chairmanship 2023*.

Richardson, C., Rymer, J. R., Mines, C., Cullen, A., & Whittaker, D. (2014). *New development platforms emerge for customer-facing applications*.

Roslan, A., & Smialek, M. (2023). Comparative Analysis of Low-Code Computation Systems. *Preprints of Position Papers of the 18th Conference on Computer Science and Intelligence Systems*, 103–110.

Sahay, A., Indamutsa, A., Ruscio, D. Di, & Pierantonio, A. (2020). Supporting the understanding and comparison of low-code development platforms. *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. https://doi.org/10.1109/SEAA51224.2020.00036

SAP. (n.d.-a). *SAP Build Apps: Build apps quickly with low-code*. SAP. Retrieved August 9, 2023, from https://www.sap.com/products/technology-platform/low-code-app-builder.html

SAP. (n.d.-b). *SAP Build low-code app development and automation*. SAP. Retrieved August 9, 2023, from https://www.sap.com/products/technology-platform/low-code.html

SAP. (n.d.-c). *SAP Build Process Automation*. SAP. Retrieved August 9, 2023, from https://www.sap.com/products/technology-platform/process-automation.html

SAP. (n.d.-d). *SAP Build WorkZone: Build and customize business sites visually*. SAP. Retrieved August 10, 2023, from https://www.sap.com/products/technology-platform/workzone.html

Tulchynska, S., Popelo, O., Vovk, O., Dergaliuk, B., Kreidych, I., & Tkachenko, T. (2021). The Resource Supply of Innovation and Investment Strategies of the Microeconomic Systems Modernization in the Conditions of Digitalization. *WSEAS TRANSACTIONS ON ENVIRONMENT AND DEVELOPMENT*, *17*, 819–828. https://doi.org/10.37394/232015.2021.17.77

Vincent, P., Iijima, K., Driver, M., Wong, J., & Natis, Y. (2019). *Magic quadrant for enterprise low-code application platforms*.

Vincent, P., Iijima, K., Leow, A., West, M., & Matvitskyy, O. (2022). *Magic Quadrant for Enterprise Low-Code Application Platforms*. https://www.gartner.com/doc/reprints?id=1-2C8VSOAH&ct=230113&st=sb

Wen, H., Zhong, Q., & Lee, C.-C. (2022). Digitalization, competition strategy and corporate innovation: Evidence from Chinese manufacturing listed companies. *International Review of Financial Analysis*, *82*, 102166. https://doi.org/10.1016/j.irfa.2022.102166

Appendix

The interviews are redacted in as to not include company names. Furthermore, certain parts of personal details are also left out of the interview texts. The interviews are also cut to only include the relevant parts to the report. Greetings, introductions and side-remarks are not included.

Interview A - SAP

Date: 31-05-2023

<u>What are the strong points of SAP Build? In other words, what specific features are you focusing on?</u>
SAP Build's strong point is the end-to-end process view, that's the unique value proposition that we have. For each feature, there is probably a competitor who does that specific feature better. But if you look at the process as a whole, competitors can't offer the same holistic approach as SAP Build can. To emphasize this, you can also look at how SAP Build evolved. SAP Build Apps evolved from AppGyver, Process Automation evolved from Workflow Management plus Intelligent Robotics Process Automation. WorkZone comes from Launchpad. We're putting all of these features together to provide the entire process view.

> <u>So, are you targeting specific business models or companies which really need this entire process view?</u>
> *To be frank, SAP targets everything more or less. That's also why it's more focused on process automation, because everything links back to this. So, we're not only targeting the large companies who can easily see the value in end-to-end solutions, but also the small and medium enterprises, whom can sometimes be hesitant to try this. They usually don't have an IT department, or already have software set-up to deal with their websites, so they don't need Build Apps or WorkZone for example. For these companies it's more about getting a foot in the door, cracking that small-mindedness, and showing them that a small automation in one of their departments works. Then the other parts of the company will see the value in it and want it as well. That's how it often goes, a small, single solution results in the entire process moving to the end-to-end solutions SAP Build can provide. This is also why SAP Build is part of the Business ByDesign Cloud ERP license. It's automatically included in the transition to cloud. So no, we don't focus on one business model or a specific customer group. What we do do, is create specific templates or scenarios for lines of businesses. For example, for procurement or for sales. For industry cases we mostly look towards our partners, such as Accenture, to onboard the customers and supply these industry customers with solutions.*
> *When we are planning, we don't even have the perspective of 'let's focus on the HR or manufacturing customers'. Sometimes Build Apps might seem like it is more for small customers, but we don't do that. We enable it and make it accessible for all. Once it's enabled, it's enabled for all use-cases, not only for IT personnel for example. We're also trying to have minimal commercial extra for implementing SAP Build, that's again, why it's a part of Cloud ERP license and mostly included in the big s4 transformation. LCNC generally works best within business units, where it starts with a small scope and naturally expands afterwards.*

What are the future plans for SAP Build, what are you focusing on right now?

We're really only focusing on two things: offering end-to-end build scenarios and increasing the implementation with other SAP products. So, we focus on creating templates, build scenarios and so on, for specific lines of businesses. The implementation with other SAP products is also a big focus now. SuccessFactor and SAP Build can already be used together, but we need to make it more flexible and easier for customer to integrate it into their existing SAP system/products.

> A question that sort of links into this, is there any focus on the integration of large language models or generative AI such as Chat GPT?
>
> *From the SAP Build apps perspective, we're looking into it. But there are no concrete plans for this year. This year already has capacity issues regarding the plans, so not this year. We can of course already integrate some of these models using external API's, but the question then becomes about data security. So external API's can be used to integrate it, but it is not realistic for businesses to utilize it for their business processes yet, due to the security issues. There is an IBM Watson and SAP partnership which will probably come to fruition next year, when we will probably have some native integration, meaning they will be secure with their data.*
>
> *The IBM partnership will lead to having some conversational AI integration. They are looking into it, and we already have a playground where they have a sales scenario up and running already, something regarding supplier onboarding. Focused on asking the chatbot to add more supplier information which it can gather from the web. But I'm not aware of any concrete use cases as of yet due to the security issues, as well as there not being real separated data sets from the rest of the world where the models can safely train on an provide insights which are not publicly available. So, the business value is not there yet. There is no quality insurance since it's only publicly available data that it can provide.*
>
> *Also, the AI discussions and planning are handled at a level above SAP Build. AI is being integrated at the top level of SAP, so SAP Build itself does not really decide this.*

Interview B – Citizen developer financial institution

Date: 04-09-2024 --- Translated Dutch → English

How did you get into LCNC development? Was it something you really pursued?

I had never really heard of Low-code No-code or anything similar to that during my studies. I actually studied a lot of different things, ranging from infrastructure to Hotel Management. I eventually decided that all that was not for me and started an IT traineeship program since I have always had an interest in computers, programming, and that kind of stuff. Basically, the digital world has always interested me, so I decided to change course and see if a traineeship in IT was something for me. This is how I got to my job at ***[Financial institution]****, since the company where I did my traineeship was also a secondment agency. I got educated in general programming, gained IT and business skills and was recommended to* ***[Financial institution]****, where they wanted me to join a newly formed team based around basically rebuilding the entire software of the company in steps using Appian. So no I didn't really pursue LCNC at all, just pivoted my way into IT and fell into it after that.*

> How long have you been working with **[Financial institution]** and with Appian?
Not that long yet, almost three months now. Starting to get into the groove of things now, which is nice. I of course had a bit of a head-start because once it was known that I would be working for ***[Financial institution]****, I had some preparation time to familiarize myself with Appian and the general work environment.*

> Do you feel like you could already start development yourself after 3 months? What position do you think you're in yourself right now?
Of course, I'd like to think that I am ready to solo develop something right now, but I'm not too sure yet. At ***[Financial institution]*** *there is a very iterative process for development, where every developer works on one part of the new application, has deadlines every week or every few weeks and then tests the work that has been done. Up until now I have mostly worked along people, watching them work and learning from them. I have done testing work, both backend and frontend and now I'm really learning how to develop something in Appian myself.*

> Do you think that this is a good way of learning how to use LCNC, or would some other ways of learning have been more effective in your opinion?
I think because I had no real previous experience with LCNC development at all, this is a very effective way of learning every aspect of the development process. I know that some people who started a few weeks before me are already at the actual development phase, and even their deadlines and development goals are tailored to them specifically. Some people found the pressure to deliver too high at the start, so the deadlines or projects that they worked on have been scaled back. I think this is a very good way of doing things. Not holding back, see where your limits are regarding developing an application or parts thereof, but also not skimping on the actual basics at the start.

> Are there any things that you yourself or other citizen developers who just started have noticed with LCNC which you did not expect, or are there any obvious drawbacks you can see?
Well, it might be a bit early for me to say, because I haven't really started to develop myself yet, at least not on the scale that I will be later down the line hopefully. But there is a definite structure in which you should work with LCNC. I know that it is also practical to work in a structured way with 'regular' application development, but it feels like there is a structured way in which to work, that not only works best, but sometimes is the only way in which it works, if you get what I mean. For us, who are just starting to use Appian, it is an easy thing to learn, because we learned this from the basics upwards. But I've heard actual developers sometimes struggle with it, that they can't do something in the way that are used to or the way

they prefer. Basically, the customizability of some aspects can be low, which is a benefit for me, since I'm just starting out, and I'm learning to do it this way, as are the other citizen developers, but for the professional developers, this can be quite bothersome sometimes. I think that later down the line because there is less customizability, there will also be less surprised or 'spaghetti' which needs to be unraveled, because basically everyone will work in the same way.

<u>What does the team make-up look like. Are there more professional developers or does it mainly consist of citizen developers?</u>

There are quite a few developers who are used to 'regular' development. This is partly because the new development with Appian will basically overhaul the entire enterprise software that ***[Financial institution]*** *is currently working with. They have the knowledge of the current, so they can translate it into the new version more easily. The team is expanding though, and there are quite a few people who are not necessarily 'citizen developers', but who have no real coding or programming knowledge, just some knowledge of IT in general.*

> <u>What is the general opinion of LCNC in your team?</u>
I can't speak for everyone of course, but most people I speak to are generally happy with it. There are of course some hurdles to overcome, but it feels like an effective way to develop an entire new IT ecosystem. I'm starting to get the hang of things, and I think it would've taken me much longer to start actual beneficial development if we weren't using LCNC. I think that is what most people feel. Of course, this is biased, because only some people were transferred from the old system to the new system. There are a lot of new hires as well, who were hired to actually do the work with Appian. But yea, overall, I'd say most people are pleased with how it is currently going.

Interview C - Neptune

Date: 08-09-2024 --- Translated Dutch → English

How much would it cost to license Neptune, what is the general order of cost?

Well, it depends highly on what is needed. In your case, for your internship, I would say it costs nothing. The free trial, I think, is more than enough for your research and for learning how Neptune works. You can follow guides in conjunction with this free version. Now, if you wanted to publish apps and create them more systematically, then the free version probably wouldn't be enough, and there are multiple packages we offer with different costs, depending on the required versions and developers.

> Thank you. What would be the cost for the SAP Edition of Neptune for a single developer?
>
> *The cost of Neptune can be structured a bit different than how other platforms. For a single developer at Neptune there is a cost of €102. €102 per developer, but at Neptune you pay for the entire DXP runtime, with each additional developer costing €102. For a DXP runtime, which includes everything including support, it would set you back €12.000. So, for the entire Neptune, SAP edition and the Open edition, it would come out to €24.000. €12.000 for the SAP edition runtime and €12.000 for the Open edition yearly with €102 additional costs per user.*

What is included in the licensing of the entire DXP runtime?

Well, there is the entire ecosystem basically. It includes support when running into issues, basically support on the development side of things, but also on the setup side. There are other ways in which Neptune can be licensed. Neptune can also create the desired enterprise software system for you, including applications, but this will of course cost more. I recall that we worked with Accenture in this way with Blueworx, in New Zealand.