Cigref
SUCCEED
WITH DIGITAL

New Low Code/No Code development practices

Unlocking value by controlling risks

December 2022

Cigref

New Low Code/No Code development practices

Unlocking value by controlling risks

December 2022

OVERVIEW

The use of "No Code, Low Code" platforms has been growing in recent years. Since they do not require any knowledge of IT development, they question the functions of the IT departments as well as their links with the business units. Do these platforms give the company's business units the possibility of doing without the technical expertise of the IT department to develop applications and transform their own processes? Or should it be seen as an opportunity for IT departments to work in collaboration with the business units, providing them with solutions that are conducive to their development with more autonomy and thus limiting shadow IT (i.e. the development of digital solutions by the business unit without control by the IT department)?

The observation shared during discussions in the Cigref working group is that these approaches democratise IT development and introduce the notion of a "citizen" developer (someone who is not an IT expert). Indeed, such platforms can be used without any in-depth knowledge of IT development, which increases the autonomy and agility of the business units and thus meets the need for rapid time-to-market. These Low Code or No Code platforms can also meet the needs of IT departments themselves for certain types of activity, enabling them to increase the productivity of "professional" developers and, from time to time, to make up for the increasingly prevalent skills shortage in IT development. However, to get the real value out of these new Low Code/No Code practices, it is essential to define a framework in partnership with the business units. To achieve this, it is necessary to implement governance to monitor and administer the use of these platforms, to define eligibility criteria and good practices for "citizen developers", to adapt technical architectures, and also to set up a centre of expertise dedicated to Low Code/No Code practices within the IT teams.

The main lessons learned from the working group's reflections are:

- Low Code/No Code **approaches are real opportunities, including for the IT department itself**: in a context of financial constraints, they can enable low-cost development and establish a formal operating framework with the business units, which is lacking, by definition, in shadow IT practices.
- **To benefit from the best conditions for success, the rules involve standardising these practices in a creative dynamic with the business units**: for example, with the development of a decision tree that structures the approach and makes it possible to analyse the eligibility criteria of the platforms, or with strong support for the business unit around the issues of security and code quality, which is adapted to the target populations (citizen developers or business units with development expertise).
- **Urban planning issues also arise in this context**: the IT department has a shared responsibility with the business units in choosing to open APIs and databases in a secure and optimised manner.

Depending on the strategies adopted, Low Code/No Code platforms can be an opportunity for business performance and innovation, but also a vector for cultural transformation in organisations.

ACKNOWLEDGEMENTS

We would like to thank **Claire WAAST-RICHARD**, Data and Digital Director at **ENEDIS**, who steered this study as well as all those who participated and contributed to this Cigref working group:

Christophe AUBIGNAC – RENAULT
Philippe AUBOURG – BNP PARIBAS
Laurent AVET – BNP PARIBAS
Emmanuel BATT – Groupe SEB
Grégory BERTHELOT – MAIF
Thierry BEY – STELLANTIS
Éric BLAWAT – CEA
Jérôme BOULLOT – HAGER GROUP
Claude BOUQUET – ENEDIS
Mohamed Ikbal BOUZGAROU – Air France KLM
Rémi BUISSON – FONDATION DE FRANCE
Sylvain CIONNEAU – SAVENCIA
Jérémy D'AURIA – ENEDIS
Yann DAUDIN – PÔLE EMPLOI
Jérôme DE GALLÉ – AXA
Joël DIEHI – MALAKOFF HUMANIS
My-Phuong DULMAN – BANQUE DE FRANCE
Stanislas DUTHIER – GROUPE ROCHER
Mariam EL-FANIDI – EDF
Pierre-Xavier FOUILLÉ– NAVAL GROUP
Vincent GALINIER – AIRBUS
Matthieu GARCIA – Laboratoires PIERRE FABRE
Jeremy GIBBONS – AIR LIQUIDE
Frédéric GIGAN – PÔLE EMPLOI
Guillaume GIROU – Groupe BPCE
Christophe GORGEOT – GPT LES MOUSQUETAIRES
Stéphane GRAS – AIR FRANCE KLM
Abdelnour GUEMACHE – SCOR
Stéphane JACQUOT – SOCOTEC
Jean-Christophe LALANNE – AIR FRANCE KLM
Hervé LAGASSIE – GROUPE BPCE
Laurent LE SAOUT – ENEDIS
Olivier LÉAL – BANQUE DE FRANCE
Stéphane LOPEZ – iMSA
Dominique MEUNIER – MAIF
Amancio MONTEIRO – MALAKOFF HUMANIS
TuChu NGO – EDENRED
Sébastien PEANNE – SCOR
Cédric PETIT – EDF
Frédéric RIGA – GETLINK
Marc-Michel STACK – BNP PARIBAS
Tristan TARBÉ – GROUPE SAVENCIA
Caroline TRAMALLONI – AIR LIQUIDE
Francis VALLET – AIR FRANCE KLM
Laurent VERHOEST – TRANSDEV GROUP
Samia VERMOET – PÔLE EMPLOI
Julien VERRIER – SCOR
Karen WILLMS – Groupe SEB
Chris WOODROW – MAIF

This document was created and written by **Flora FISCHER**, Mission Director at Cigref with the contribution of the work's manager and participants.

We would also like to thank all of the participants whose input guided our study:

- **Alain FAURÉ** and **Sylvain FAGNENT**, Manager, No/Low Code specialist at **OCTO TECHNOLOGY**
- **Christophe GORGEOT**, DEV FACTORY Section Manager at **STIME (LES MOUSQUETAIRES GROUP**)
- **Éric BLAWAT**, MOE Project Manager at **CEA**
- **Matthieu GARCIA**, Company Architect at **PIERRE FABRE**
- **Benoit LOCU**, Head of Enedis Innovation Labs at **ENEDIS**
- **Caroline TRAMALLONI**, Group CIO Office, Enterprise Architect at **AIR LIQUIDE**
- **Jean-Pierre PITRAU**, IT4IT and FastTrack Dev Manager and **Laurent LETELLIER**, IT Product Manager for Low Code/No Code solutions at **AIR FRANCE KLM**

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF INSERTS

INTRODUCTION

Although Low Code and No Code platforms are not new to large organisations, they have become increasingly attractive to teams over the last few years, particularly as some of them are offered "free of charge" and sometimes without access constraints with bundled offers. The market for offerings is growing and the functional coverage is increasingly comprehensive, ranging from the implementation of workflows to data processing and the development of applications or web applications. With Low Code or No Code, these offers are becoming increasingly attractive to businesses that use them to meet a specific need for autonomy or speed of execution. They also appeal to "professional" developers who, thanks to the automation or abstraction of purely technical tasks that Low Code platforms allow, will be able to concentrate more on the functional aspect of applications. Between flexibility, speed of development and production and a cost that (at first glance) seems low, these platforms seem to offer an answer to the bottleneck of requests for the design of small applications, which many organisations face due to lack of time and resources.

However, it is clear that these platforms, despite all the opportunities they promise, sometimes reinforce shadow IT, i.e. the development of digital solutions without the involvement of the IT department, which is both problematic in terms of governance and risky in terms of IT system security. These shadow IT approaches will inevitably pose problems of operability, maintainability, security, application redundancy, and even cost on "simple" business applications developed in Low Code/No Code. These can become very complex or even critical over time, and ultimately cost more than a more traditionally developed application.

Shadow IT is a reality that can be accentuated by Low Code/No Code if it is not supervised. The implementation of clear governance and support as close as possible to the business units are steps that seem unavoidable for the successful deployment of these platforms, especially since, according to a Gartner study, "*there will be four times as many citizen developers as professional developers in 2023 [...] and there will no longer be enough (IT) professionals to meet the needs of companies*." [1]

IT departments have become aware of these elements. But, in practice, how can we share with the business units the **biases or grey areas of** Low Code/No Code **practices** which, at first sight, do not require any development knowledge? **How can we organise, lead and support** the deployment of these practices within the business units **while guaranteeing the right level of security and optimum technical support?**

In a context where the demand for in-house application design continues to grow, could the democratisation of these Low Code or No Code uses be an answer to the digital transformation of business units?

[1] https://www.lemagit.fr/actualites/252490733/Low-code-no-code-les-geants-du-cloud-bousculent-les-lignes-du-marche

1 FINDINGS

1.1 POSITIONING OF IT DEPARTMENTS IN RELATION TO THE LOW CODE/NO CODE APPROACH

More and more IT departments are asking themselves what to do with Low Code/No Code platforms. Should their access be blocked while we study the best ways to avoid shadow IT and the associated risks? Or should we experiment and, in this case, ask ourselves how to define the uses, how to segment them, how to support the business units, which platforms to choose, how to monitor the development of the business units over time? Many IT departments argue that it is not so much a question of controlling and restricting usage as it is of orchestrating it to derive value and manage the risks.

1.2 DEFINITIONS AND FIRST APPROACHES

The major difference between **traditional development** and Low Code **or** No Code **approaches** is that Low Code/No Code platforms provide high level, ready-to-use unitary components that the citizen developer assembles according to the functionalities they want to implement. The approach is very functional and the technical complexity of development is hidden in the reusable components. These components are therefore accessible to non-specialists, which limits costs, lowers the level of technicality required to develop an application, contributes to business agility and responds to the lack of developers thanks to other profiles (which will have to be trained, however, to keep the applications simple). Traditional development, on the other hand, requires writing code explicitly. It allows complex needs to be understood, but also to optimise the code so that it is, for example, more efficient in its consumption of drives or memory (and therefore energy), more easily exploitable or maintainable over time, including when security flaws need to be dealt with.

The No Code promise is to provide intuitive interfaces that are accessible to anyone without knowledge of the code. No Code is particularly suitable for prototyping, MVP (minimum viable product), because it makes it possible to quickly design small applications, create for example very simple web pages, integrate data in a dynamic way or external applications via APIs or quickly publish on a website or an application.

Low Code is quite different. It makes it possible to design enterprise applications that have a sustainability objective. Its functionalities are richer and more complex, but not business critical. They allow, for example, the construction of pages or the configuration of workflows, the integration of applications with the company's IS, and the management of environment configuration. Low Code requires greater IT skills and training, although the Low Code/No Code boundary is blurring as No Code tools offer increasingly advanced design options.

NO CODE	
What	**Intuitive** tools accessible to all
Why	• Prototypes / MVP • Extended office automation
For whom	**Citizen Developer**
Solutions	Glide, Adalo, Appsheet, Weebly, Dropsource, Draftbit, Zapier, Airtable, Caspio, Bubble, BettyBlocks ..
Distinctive features	+ **Easy to use** + **Quick results**

LOW CODE	
What	Tools requiring IT **skills** and **training**
Why	Allowing to **create** and **deploy enterprise applications**
For whom	**Model Developer**
Solutions	Mendix, Outsystems, PowerApps, Appian, Salesforce, Pega...
Distinctive features	+ **Versatile** + **long term solution**

Figure 1: Octo Technology, "Two types of tools for different uses and user-developers", intervention at the Cigref "Low Code, No Code" working group, January 2022

It is important to point out that **the simpler the tools are, the more they will constrain the uses** and types of applications possible. Based on this observation, it is therefore essential to know how to address Low Code/No Code solutions to the appropriate populations, according to their different needs:

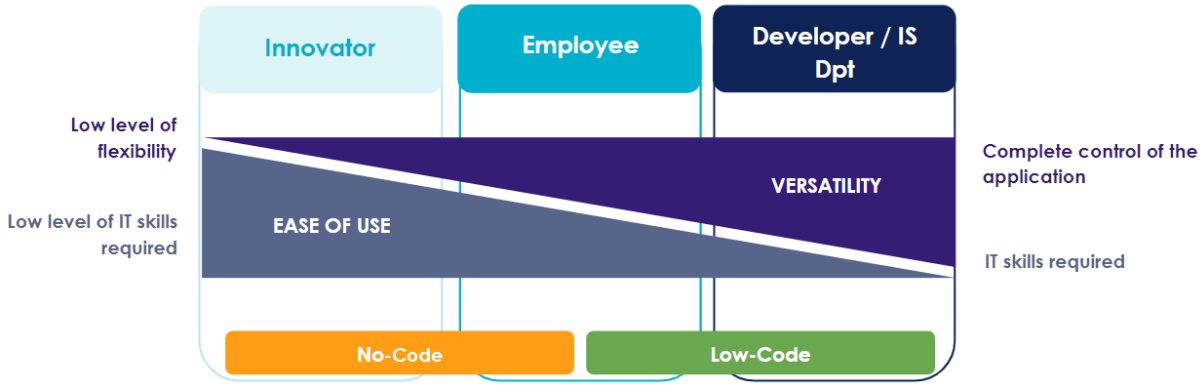

Figure 2: Octo Technology, "Different compromises for different populations", presentation within the framework of the Cigref "Low Code, No Code" working group, January 2022.

According to this Octo Technology model, three types of populations are concerned by these solutions: The entrepreneur/innovator, the employees and the developers and IT teams.

For example, **the entrepreneur-innovator** wants to create a **prototype** to **validate hypotheses about a product**, develop an MVP to **start an activity, or create an application for a specific event**. With No Code, this entrepreneur/innovator profile will have tools that are easy to use, focused on a specific problem (web or mobile site, data management), with the possibility of assembling different bricks by speciality. While this usage gives little visibility of the model and the revenues generated, it does offer

great simplicity and speed of implementation. Moreover, going for the shortest possible time can help to acquire the necessary maturity to be able to scale up a richer solution later on, and cover wider needs.

Employees, for their part, use these platforms perhaps more occasionally, but with **quality and durability requirements** that require a more detailed use of these platforms. The most common use cases are:

- The automation of low value-added and time-consuming operational tasks (currently often done via office automation);
- The creation of tools to manage business processes to facilitate and simplify the contribution of the different participants;
- The creation of tools that facilitate operations (such as mobile support or input applications).

Finally, **for the IT department's teams**, the use of Low Code is also a new way of responding to the needs of business units or to their own needs: for example, it allows them to reduce the time of a project by having an application factory, to model data or to lighten the operation by using platform solutions.

Once the needs have been identified and the target populations have been defined, what framework can the IT department put in place to optimise the implementation and security of these initiatives?

2 GOVERNANCE AND ORGANISATION TO MONITOR AND MANAGE LOW CODE/NO CODE PLATFORMS

Setting up a governance structure dedicated to Low Code or No Code practices in the company is a necessary prerequisite for the successful deployment of these platforms. This governance must enable action to be taken at various levels, as much organisational as technical and managerial. This covers, for example, the definition of selection criteria for Low Code/No Code solutions with businesses, the development of technical and architectural requirements, the taking into account of skill requirements, the risks of maintainability and security of applications, and the support of the businesses.

2.1 THE OBJECTIVES OF A LOW CODE/NO CODE GOVERNANCE

From the IT department's point of view, governance must enable the solutions produced on these platforms to be controlled, practices to be harmonised and the management of Low-Code/No-Code developments to be optimised, by:

- Controlling costs, especially due to freemium models[2];
- Avoiding the multiplication of products and streamlining information;
- Offering a structured support service (PQSR, templates, training, tutorials, etc.);
- Controlling deployments, including identifying successful applications and bringing them under control;
- Mastering the urbanisation of solutions;
- Ensuring the selection and then the pooling and visibility of the use cases developed;
- Implementing rules to meet business needs for autonomy or speed of execution while satisfying IT requirements (maintainability, security, GDPR compliance, etc.);
- Assisting business units and communicating on the possible scope of autonomy of the business unit, in particular concerning technical aspects.

This governance often requires the setting up of a **centre of expertise** to address the issue of the use of Low Code/No Code platforms. This makes it possible to **prepare calls for tender** in order to make an informed choice on integrators or publishers. Several points of vigilance may then be qualified such as the intrinsic quality of the code, the types of connectors proposed and their possible reversibility.

2.2 MANAGEMENT OF THE APPROACH

In terms of management, several positions can be defended: it may be appropriate to give more autonomy to the business units as long as the projects remain small-scale and non-structural, but if the aim is to systematise the use of these platforms, it is important to clearly define the roles in each business unit and entity, to develop a joint IT-business unit management, even if the IT department can take control of governance, by structuring it by business unit or entity.

[2] *Freemium*: A business model that offers a free service aimed at attracting the largest number of users but with more advanced complementary services for a fee.

Having a dedicated entity, on the development side, or a specialist interfacing with the business units seems essential to remain as close as possible to the challenges and expectations of the business units. Furthermore, developing a practice of analysing the value of these initiatives and entering into a logic of capitalisation and pooling of use cases and success stories, via a "market place" for example, will make the approach visible and attractive.

In the example of one organisation (Pierre Fabre), three roles were defined to work together in order to manage the requirement of the business units:

- The Business Process Owner **(BPO)**: a business specialist who is already part of the business units, their mission is to better qualify requests, to manage the business urgency and its need. They are the cross-functional specialist for processes and data.
- The Business Unit IS Coordinator **(CSIB)**: guarantees the capacity to implement projects on the business side, and ensures consolidated reporting on IS activities for the BU.
- The Business Relationship Managers **(BRMs)**: distributed by area (R&D, sales & marketing, operations, HR/Finance, etc.), they manage business demand and frame the IS requirements for their functional area. Cross-BRM committees can be organised on a monthly basis to strengthen overall coherence.

Feedback from Pierre Fabre
Business support for small app projects

Pierre Fabre has set up a specific entity, called Digital Acceleration and Information System (DAIS), the result of the collaboration of a process & organisation entity, a digital entity and a traditional IT department. Its objective is to provide a total service offering that integrates the governance of business processes, data management, and the implementation and maintenance of solutions, in particular Low Code.

One of the solution modes that was opened up to business units in 2018 is small Apps. On this subject, the business unit is given autonomy in terms of delivery but within a controlled framework. The DAIS will then:

- Ensure that the service does not already exist or is under construction in the DAIS catalogue;
- Reference the solution;
- Ensure compliance with safety rules, regulations and group standards;
- Define and maintain reference architectures and solutions;
- Provide secure, monitored and backed-up platforms;
- Manage the integration with the IS (APIs/Interfaces);
- Provide guidelines and technical expertise.

The basis of this approach is demand management (and therefore dialogue and exchange between the 3 participants: BRM, CSIB, BPO) which helps guide the business units towards this type of solution.

Matthieu Garcia, *Business Architect, Pierre Fabre*

2.3 ELIGIBILITY CRITERIA

The issue of governance arises from the platform selection phase, based on eligibility criteria that will be defined according to the user's scope and needs. It may be appropriate to start by defining a **scope of use** and **exclusion** via a **decision tree**. This can be built with the business units, as in the example below:

Feedback from STIME (Groupe Les Mousquetaires)
Low Code/No Code decision tree

The Dev Factory teams (the division managing DotNet/React and AppSheet/PowerApps applications) have worked on a decision tree to validate whether an application is feasible with AppSheet or with PowerApps, the two most widely used solutions in the group (if it is not feasible with either of the two solutions, traditional development will be necessary). The first stage is based on the validation of the Design Authority, a decision-making body bringing together developers, users, architects and security teams. Its role is to:

- Review applications created over the period with the aim of avoiding duplication and detecting applications that can be focused on;
- Review and accept application architecture patterns combining No Code platform and interaction with the IS;
- Review requests to make data sources from the IS available via API.

A questionnaire is also issued to the business units. It includes typical questions such as:

- Is the application strategic, what is the target user base, what are the performance requirements?
- What is the level of UI/UX complexity?
- How long does the application last, etc.?

Based on the responses, recommendations will be made. If an application is selected, depending on the type of project and audience, different bodies will be mobilised: an architecture committee and/or a governance committee. Then, it will be decided whether a training plan should be launched, or whether deployment can be started directly. The application is then maintained and monitored by the Dev Factory team, which may report to the governance committees.

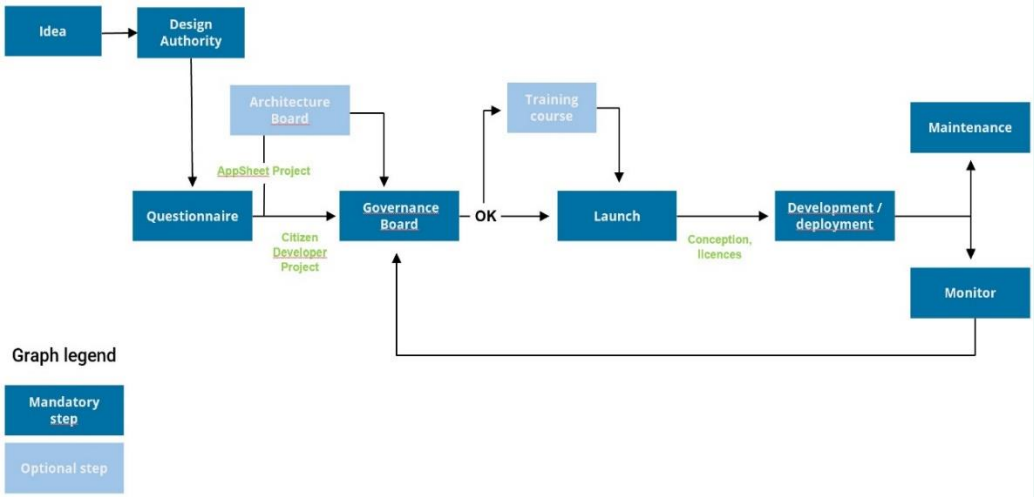

Figure 3: Low Code/No Code decision tree (STIME)

Christophe Gorgeot, *DEV FACTORY Section Manager, STIME (Groupe Les Mousquetaires)*

Developing a **decision tree** allows you to assess the relevance of the choice of a Low Code/No Code platform or of a specific development, and to address for example the questions that arise regarding the strategic aspect of the application, the number of users involved, the performance needs, the level of complexity of the UI/UX[3], or the life span of the application.

Eligibility criteria may focus in order of priority on:

- **The cost of licences** and their **model**: the predictability of costs with regard to the intended users is not always easy. It is also necessary to clearly define the scope of use and to question the choice of licence made according to the number of users or the number of developers. The ability of IT to administer the portfolio should also be taken into account.
- Platform **security**: a SecNumCloud certification requirement for the hosting company, the guarantee of structured and secure access to data, the possibility of auditing the IS or hosting companies, the responsiveness of the publisher's support, and the stability of the environment will be key criteria to take into account. The **level of risk** must also be assessed internally, according to the criticality of the data of the applications concerned or their level of exposure to the company's strategic data.
- **Rapid ramp-up**: it is essential that the solution can be picked up quickly by users and that the training period is short and efficient. Design UX is an essential criterion for the approach to be appreciated by the business units.
- **Sustainability and scalability** of the solution: it is important that users who develop applications via Low Code/No Code platforms ensure their sustainability over time by documenting their work as much as possible. Specific support and awareness-raising on this subject are essential.

[3] User Interface/User Experience

- **Native connectors**: these must allow rapid and optimal integration of Low Code/No Code applications into the company's IS, but also to operate in DevOps or in "offline" mode.
- Ease of application of **digital accessibility** rules: these rules must be taken into account as early as possible in the design process. Low Code/No Code platforms do not necessarily take into account in their offers this strong societal issue, which is still too little known. In addition, the regulations are evolving in this area. Accessibility is already compulsory for a certain type of organisation (Article 47 of the Law of 11 February 2005 for equal rights and opportunities, participation and citizenship of disabled people) and the European Products and Services Directive[4] will include new services in this obligation.

2.4 SUPPORT AND BUSINESS RESPONSIBILITIES

Finally, **supporting employee practices** is at the heart of governance. Some organisations have put in place a specific no-code strategy with guidelines and recommendations to ensure that good practices in cybersecurity, confidentiality and technical maintenance are followed and respected. Several companies also negotiate with their integrator or publisher to provide support in the deployment of governance and in the setting up of small training courses adapted to the profiles that will use these platforms.

Governance must be based on **education with the various stakeholders**. The IT department cannot impose its rules alone, it is necessary to put the business units in charge, as in the agility processes.

Two positions can be taken regarding the issue of business support. The IT department may choose to set up a **pool of developers for the business units**: in this case, the IT Department makes developers available to the business units and takes charge of urbanisation and risks. But this comes at a cost.

The other positioning is to **propose development "support"** by business units, without having a dedicated entity. In this case, it is important to target the populations, usually by allocating No Code for citizen developers and Low Code for professions with more expertise. The idea of proposing hackathons is good for raising awareness and providing support in a fun way.

[4] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2019.151.01.0070.01.ENG&toc=OJ:L:2019:151:TOC

Good practices in summary

In terms of governance and management of Low Code/No Code solutions, the following good practices can be retained:

- The development of a **decision tree**, co-constructed with the business units in order to identify the key eligibility criteria for Low Code/No Code solutions;
- The definition of the **scope of use** of the application in order to clarify the needs and to anticipate the necessary means and resources;
- **The mapping of** Low Code/No Code **applications** already in place and the provision of this catalogue to the business units;
- The definition of **roles**: be clear about the responsibilities of each party, for example concerning **MCO** and **security**, and include **architects** and **cybersecurity** teams **as early as possible in the processes**.
- **Business support** in order to anticipate risks and promote good practice:
  - **Share** what is being done, and have the applications developed **evaluated** by end-users;
  - Set up a **community of good practices** to capitalise on good and not so good experiences;
  - Offer **short training courses** for citizen developers;
  - Establish **a permanent dialogue with the business units** on: education, risks, the value of these platforms;
  - **Propose hackathons** around Low Code/No Code with the business unit: this type of challenge has a fun aspect and can make the support more effective.
- **Identify the skills required** in order to anticipate the mobilisation of resources and skills:
  - Create a rapid development team to support business needs;
  - Or authorise development by the business unit, when the applications are not critical.

3 CHALLENGES AND OPPORTUNITIES FOR IT

In this context, IT departments have to face various challenges in the operational phase of the deployment of Low Code/No Code platforms, whether it is a question of urban planning and architecture, cybersecurity, reversibility or cost management, but also challenges in terms of innovation and enhancement of IT-business relations.

3.1 URBAN PLANNING AND ARCHITECTURE

No Code forces us to consider simple solutions for non-critical processes: business users will therefore start projects with enthusiasm but with a risk of disappointment because the application will have limits, particularly in terms of ergonomics and functionality. The IT department must take into account this risk of requests for functional changes that sometimes cannot be resolved. This touches on the limits of off-the-shelf solutions. This raises the issue of **user design**: IT departments are called upon to make interfaces more fluid in order to improve the user experience. In this case, it may be interesting to take this parameter into account in the selection criteria for Low Code/No Code offers, or to offer usability and UX design training to the business units, or to dedicate a dedicated team to this. Another challenge for the IT department is **usage planning**, which concerns the organisation and referencing of Low Code/No Code applications in order to optimise performance and resources for each initiative.

Some IT departments note that with No Code, only the development phase is really accelerated and that little time is saved on the whole process, because of the greater or lesser complexity of connecting the flows, depending on the environment chosen. In fact, some environments require the use of web APIs and specific developments. It is sometimes necessary to define a short delivery circuit and to adapt the company's processes for these less sensitive applications.

3.2 CYBERSECURITY

IT departments have to deal with different types of risks: the **use of data** is one of the first issues. First of all, it must be ensured that the use of the data does not jeopardise its integrity, compliance or the security of a channel. In addition, some applications can sometimes be diverted from their initial use, which makes it necessary to review the entire process. Ensuring compliance with **security and confidentiality frameworks** and access permissions is a key issue in these ever-changing and evolving environments.

Cybersecurity teams already concerned with improving the security by design of traditional developments are naturally involved in the Low Code/No Code approach. These teams are comfortable with certain solutions that are in already marked and secure environments. One of the major difficulties encountered is in Low Code/No Code models that do not embed their own database, as this requires exposing the company's data and setting up APIs to secure queries.

3.3 REVERSIBILITY

One of the points that raises the most reluctance to the systematisation of Low Code/No Code approaches is the question of reversibility (or "vendor lock-in"): this means that once the subscription to an offer is terminated, the application becomes unusable. One of the solutions is then to push the teams to **document** their work **as much as possible**, like for traditional development, so that they can recover information if they have to rewrite the application in a new environment. Another possibility is the use of **standard** frameworks, or the use of open source platforms. Many companies have open source strategies that address their need for sovereignty and reduced technological dependency. They may therefore consider the Low Code/No Code approaches as interesting in this sense.

3.4 OPPORTUNITIES

Low Code/No Code approaches are also real opportunities, including for the IT department itself, if we consider, for example, that they allow **development or enrichment of development platforms at a lower cost** in a context of financial constraints, or to establish an operating framework with the business units that is more secure and standardised than shadow IT. The Low Code/No Code approach can also **serve the** organisation's data **strategy**.

The result may be a real dynamic of creativity and innovation with the business units, with a service offer that responds **more quickly** to the need for small and medium-sized applications, and the possibility for application managers to design **prototypes** more easily. Another positive dimension of these approaches is **the acculturation of the business units to IT issues**, thanks to the appetite and curiosity that these sometimes playful and creative environments arouse. The collaborative aspect of Low Code was thus highlighted by the experience of **Enedis**:

Feedback from Enedis
Low Code as a lever for data project transformation

Enedis has chosen a solution that targets data analysts in the regions to improve the upstream phase of their data project and to do so in complete security in a single, secure environment, thus limiting local shadow IT initiatives. This solution is very powerful in the preparation and cleaning of data, and offers the guarantees of an industrial and secure solution. It can be installed on-premise, and is one of the few to offer this possibility. In addition, it facilitates collaboration between different teams. Thus, the solution has improved several aspects in the execution of data projects by:

- Making a collaborative dynamic possible between data scientists at national level and data analysts in the regions;
- Reducing implementation time and optimising the use of scarce resources;
- Moving from the experimental stage to the industrial stage in a seamless and "by design" manner;

- Acculturating the business units to the controlled opening of data: the solution is linked to the company's datalake, which allows connection to BI (Business Intelligence) tools, creation of data files, APIs or internal data services.

Benoit LOCU, *Head of Enedis Innovation Labs, Enedis*

Lastly, Low Code/No Code governance makes it possible to **propose a framework** for this innovation dynamic, by involving all participants in a standardised logic, by setting up decision trees, and by animating the communities of developers and citizen developers (not without challenges, since there is a high turnover in the development professions), while maintaining the objectives of security and code quality.

Good practices in summary

In terms of IS urban planning and architecture of Low Code/No Code solutions, the best practices to be retained concern the following points:

- Have a **good "APIsation" maturity** of the IS and to acculturate all participants to the stakes of data opening;
- Require a **documentation of the tools** at the same level as that of a traditional specific development in order to avoid vendor lock-in, or favour open source **approaches**;
- Define and put in place **rules for usage planning** or even **usage design** in the face of No Code multiplication: this should make it possible to frame the evolution of functionalities in order to prevent the application from becoming unusable or non-user-friendly. These rules should also prevent corporate data from being ingested unchecked by Low Code/No Code solutions;
- **Check annually the catalogue of applications** and their possible optimisation;
- Articulate the use of these platforms with the organisation's data **strategy**.

4 GUIDELINES FOR CITIZEN DEVELOPERS AND IT TEAMS

The following guide, based on the reflections of a participating organisation, was enriched by the members of the working group. It presents best practices for citizen developers on the one hand, and IT teams on the other.

4.1 FOR CITIZEN DEVELOPERS

1. **Using the tools on defined perimeters**:
   - The applications developed should be of low complexity and not concern critical business areas. They can be used, for example, for communication, one-off events, internal processes, etc.
   - Applications should be developed by a single department or team.
2. **Ensuring ownership of the application**: ownership will either be that of the citizen developer or that of another person who chooses to be responsible for and manage the entire life cycle of the application. If no one volunteers to monitor and maintain the application, it will have to be decommissioned.
3. **Defining when the application becomes critical**: if the functional scope increases and the solution concerns a larger number of users, if the application blocks processes or is unavailable, it is necessary to delegate its entire management to IT.
4. **Using existing APIs to feed the applications to be developed**: Secure access to data is best achieved using existing APIs: this ensures secure access to qualified data (customers, contracts, etc.) and allows the application to guarantee data quality and consistency.
5. **Prohibiting redundancy of the functionalities already present in a government application:** The applications developed must not duplicate or replace existing applications or create bridges with applications already in place (refer to the organisation's application catalogue). They should be developed with the aim of facilitating, simplifying or accelerating data capture, or reducing tedious communication between teams or departments.
6. **Sharing feedback** within the framework of a community of citizen developers by setting up a market place, for example.
7. **Arbitrating the maintenance of an application according to the ROI or the evolution of the application's criticality.**

4.2 FOR IT TEAMS

1. **Defining secure areas**: a strong partnership between IT and the business units is needed to avoid any negative impacts of shadow IT. Boundaries and a framework need to be defined by

offering support from professional developers, by developing strict IT oversight and governance prior to release, and by identifying what is entirely outside the scope of citizen developers and needs to be managed within the IT portfolio and by IT departments.

2. **Having identified use cases** such as form design, data collection application design, business process orchestration or workflows. If the business units want to develop other applications, that are outside of this list or too complex, it is necessary to consult IT.
3. **Making maximum use of what already exists in the** cloud: authentication, authorisation and ownership processes should be reused by citizen developers, to avoid re-developing these paths. The use of features provided by the cloud (security authentication, separation of user rights) gives more possibilities to apply governance and security rules (rule-based permissions, etc.).
4. **Using APIs**: in order to interface systems and transfer data into their application, citizen developers should use available existing APIs, by consulting the organisation's API catalogue. If no API is available, IT management should be contacted to assess how best to access the required data.
5. **Defining roles and supporting the** citizen developer **on security**: when it is not possible to use an API, it is essential to secure the data by implementing access controls, configuring the application, and defining roles and responsibilities. These tasks require the expertise of the IT department.
6. **Respecting the classification of data that is imposed internally** (and that must be known by all) in order to respect the desired level of security. Audits may be carried out afterwards.

CONCLUSION

The Low Code/No Code movement is still in its infancy. It is a positive symptom of the increasing digitalisation of the company's business units, and of an evolution of practices among developers. Its users, whether citizen developers or professionals, increasingly want access to environments similar to those they experience in the consumer computing world. The response offered by the use of these platforms to the congestion of IT departments and the empowerment of the business units is a godsend, but it is necessary to orchestrate and control these new environments throughout their entire process.

The relationship between the IT department and the business units is at the heart of a specific governance system, making it possible to channel and arbitrate the choice of these platforms according to numerous criteria relating to user needs, urbanisation or security, as well as the definition of roles and responsibilities.

Certain methodological tools are essential to mobilise, such as the decision tree or the questionnaire co-constructed with the business units. They help to identify the key eligibility criteria for Low Code/No Code solutions, but also to promote the acculturation of the business units to the IT challenges and their responsibility in this area.

The years to come will show whether the lack of development skills will persist and lead companies to voluntarily favour this type of approach to compensate for the shortage of developers, or whether Low Code/No Code will not, itself, give rise to new vocations enabling these skills to emerge within organisations.

Cigref
SUCCEED
WITH DIGITAL

Achieving digital success to help promote the economic growth and competitiveness of its members, who are major French corporations and public administrations, and users of digital solutions and services
Cigref is a network of major French corporations and public administrations set up with a view to developing its members' capability to acquire and master digital technology. It is a unifying player in the digital society, thanks to its high-quality thinking and the extent to which it represents its members. Cigref is a not-for-profit body in accordance with the French law of 1901, created in 1970.
To achieve its mission, Cigref counts on three business units, which make it unique.

Belonging
Cigref speaks with one voice on behalf of major French corporations and public administrations on the subject of digital technology. Its members share their experiences of the use of technology in working groups in order to elicit best practices.

Intelligence
Cigref takes part in group discussions of the economic and societal issues raised by information technologies. Founded nearly 50 years ago, making it one of the oldest digital associations in France, it draws its legitimacy from both its history and its understanding of technical topics, giving it a solid platform of skills and know-how, the foundation stones of digital technology.

Influence
Cigref ensures that its member companies' legitimate interests are known and respected. As an independent forum in which practitioners and actors can discuss and create, it is a benchmark recognised by its whole ecosystem.

www.cigref.fr
21 av. de Messine, 75008 Paris
+33 1 56 59 70 00
cigref@cigref.fr