Universiteit Leiden

ICT in Business and the Public Sector

A global rollout methodology for low-code /no-code platforms for a business with high local autonomy

Name: Yasmin Dabbas
Student-no: s2733463

Date: 18/07/2022

1st supervisor: Prof.dr.ir Joost Visser
2nd supervisor: Dr. G. Guus Ramackers

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Background: Low-code/no-code platforms promise to enable organisations to adopt technological improvements quickly and effectively. They enable a variety of users, including non-technical people, to create solutions that move away from traditional systems and digitize paper-based operations. In some cases, they offer ready-to-use templates that may be deployed straightaway to the users via mobile applications and tablets. These platforms are intended to increase productivity while also lowering risks associated with manual operations. However, some organisations face challenges in rolling out these platforms globally, especially for multinational organisations where the local entities have high autonomy.

Aim: The aim is to provide a framework for organisations that addresses the challenges and considers the success factors for the rollout of low-code/no-code platforms.

Method: We conducted this research in collaboration with a host company, a low-code/no-code platform vendor. We performed 14 exploratory interviews with organisations who use this platform or considered using it. The interviews were conducted in order to identify the challenges and the success factors in the existing rollouts. Representatives from a variety of departments and countries took part in the interviews, including departments of innovation, IT, operations, and procurement.

Additionally, we performed a survey with 53 low-code/no-code platform end users to detect any concerns with the existing platform's utilization.

Finally, we did an evaluation of the rollout framework that we designed with the low-code/no-code platform management team to evaluate the usefulness of the framework.

Results: The outcomes of the explanatory interviews reveal a variety of factors that influence rollouts, which we categorized into three contexts: organisation, technology, and external environment. These contexts were considered in designing the rollout framework.

According to the end-user survey, the existing rollouts for the low-code/no-code platform did not generate any significant concerns. However, some of the findings contributed in the design of the rollout framework.

We evaluated the designed rollout framework and based on the feedback obtained, it was separated into two rollout frameworks that organisations could use. A simplified, light-weight framework and another framework based on essential SAFe. The suitable framework should be chosen depending on the business need and the organisation's characteristics.

Conclusions: The speed of adoption of the low-code/no-code platform is due to a number of factors other than technology of the platform. The lack of clarity in the process and responsibility, the management support, the cost and budget allocation, proper data management, a feedback procedure and support setup as well as the resource availability, are the key factors that impact the speed of the rollout.

Keywords: Low-code, no-code, digital transformation, citizen developers, rollout framework

Contents

Acknowledgement

Writing my thesis was a really challenging task for me. Without the tremendous help I received from several people, I would have struggled to focus and complete my research because of my burnout.

I want to start by expressing my sincere thanks to my supervisor, Prof.dr.ir. J.M.W. Visser, for his unconditional support and insightful comments throughout our weekly sessions. We had a lot of interesting discussions, and his academic expertise as well as his professional advice really enhanced my knowledge base.

I would like to thank Dr. Guus.J. Ramackers, my second supervisor, for his understanding and valuable advice during this research. I would also want to express my appreciation to Dr. Christoph J. Stettina MSc for his input on SAFe.

Additionally, I want to thank the host company and my supervisor in particular. Without the great support, mentoring, and encouragement, I would not have been able to undertake this journey. Working with the teams in the company provided me with a lot of knowledge.

An extra special thank you to the organisations that contributed significantly to my study. Words cannot express my gratitude for volunteering their time to take part in the study.

Finally, I want to express my gratitude to my family and friends who never stopped believing in me and pushing me to complete my thesis.

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
ART	Agile Release Train
GDPR	General Data Protection Regulation
HQ	Headquarter
KPIs	Key Performance Indicator
MVP	Minimum Viable Product
NFRs	Non-Functional Requirements
PI	Program Increment
POC	Proof of Concept
QA	Quality Assurance
RACI	Responsible, Accountable, Consulted and Informed
ROI	Return on Investment
RTE	Release Train Engineer
SAFe	Scaled Agile Framework
SPCs	Safe Program Consultants
TCO	Total Cost of Ownership
UI	User Interface
UTAUT	Unified Theory of Acceptance and Use of Technology

1. Introduction

1.1. Background

Low-code/no-code platforms provide businesses with simple development solutions that are intended to offer rapid and effective technology upgrades. They are less expensive than traditional development and do not require the same IT expertise. Figure 1 compares the creation of solutions for four cases using the traditional development using programming languages to the development of solutions using low-code platforms [1].

Enterprise	Result	Code	Low-code
US government (Affordable Care Act)	Document compliance module	100 person-months	5 person-months
British insurance provider	Agent portal	Unknown*	10 days to minimum viable product (MVP)
Call center operator	Customer-specific app	4 months	3 weeks
Spanish insurance provider	Web channel and administration system	2.7 years (estimated)	13 weeks

*The project was on the technology management backlog list for years with little hope of ever starting

Figure 1 Effort comparison of low-code development and traditional development in four cases [1]

According to a Gartner study, the adoption of low-code software development platforms is increasing at a rate of 20% annually [2]. COVID-19 contributed to this increase due to the need for remote solutions that are accessible to users and help improve the efficiency of the operation.

Automating processes using low-code/no-code platforms has a potentially lower implementation cost than a ready-made software that can be bought and used immediately or developing applications from scratch using the traditional way. Given that organisations are adopting agile because they want to increase the speed to market, low-code/no-code platforms are intended to offer a fast way to build MVPs with low-cost prototypes that promote their digital transformation and reveal the actual requirements at an early stage of the project. Building the applications has low maintenance and offers scalability across all regions.

There are different reasons for organisations to consider the low-code/no-code applications. In a survey performed with more than 3,300 IT professionals around the globe, 66% chose the primary motivation for using low-code platforms was to accelerate their digital transformation and increase responsiveness to the business, while 45% of the surveyed professionals pointed to the reduction of dependency on hard-to-hire technical skills [3].

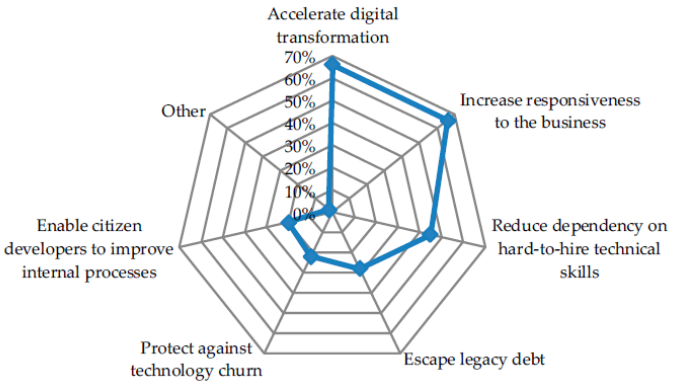


Figure 2 Reasons for using low-code development platforms [3]

Low-code/no-code platforms provide better prospects for many personas, with Citizen Developers being one of the important personas that low-code/no-code platforms target. A Citizen Developer is an employee who develops applications using technologies that are not strictly for IT use. It represents a persona rather than a title or role, and typically reports to departments other than IT [4].

Citizen development should get considerable adoption and offer value in three unique and growing markets [8]:

1. Faster traditional development: Citizen developers allow IT to dramatically accelerate the delivery of traditional applications.
2. Business innovation: enabling business teams to design their own solutions in an effective and secure manner, hence bridging the IT capacity gap.
3. Shadow IT: assisting CIOs in regaining control over software developed outside of the IT department or without its approval.

1.2. Problem Statement

Digital transformation has become a critical phase in the organisations. External elements that promote the need for digital transformation, according to Peter et al., include the introduction of new digital technologies, worldwide and intense competition, and lastly, changes in customer behaviour [17].

Digital transformation should be viewed as a whole organizational strategy rather than just transitioning from analog to digital, according to a study conducted by Mergel et al. based on 40 expert interviews from 12 countries, the majority of which were from the government sector [18]. Technology is essential for initiating transformative processes, but it is insufficient for establishing long-term benefits. Organizational culture, as well as personnel capabilities and mindset, must be considered in order for the digital transformation to be sustainable.

The current low-code/no-code implementation process began in the innovation department, where a business requirement was discovered and some manual processes needed to be automated. Low-code/no-code platforms were among the technologies proposed for resolving the problem. Multiple vendors were chosen based on various factors, and the host company was chosen to complete the prototype. The prototype was approved, and the proof-of-concept was implemented in some entities. Other obstacles and challenges arose during the process of rolling out this platform globally, slowing down the adoption speed.

When the low-code/no-code platforms are implemented, it is critical that the business need is covered and the rollout proceeds smoothly so that the advantages may be realized early in the project. The focus company in this study has a high level of local autonomy that wants to improve the speed of rolling out the low-code/no-code platform to all entities. Given the few entities that have implemented the low-code/no-code platform, the faster rollout speed of this application allows realising the benefits earlier. The company already paid for the licenses and customisations which are better to be used on a larger scale.

1.3. Research Context

We collaborated on this research with a low-code/no-code platform provider. For privacy reasons we cannot give the exact name of the company, so we will simply refer to it as "the

host company (HC)" throughout this paper. Their platform will also be referred to as "HC" to represent the low-code/no-code software that they offer.

The HC hosted the research, which focused on one of their international clients. This client is an international organisation that has multiple entities around the world where low-code/no-code platform needs to be deployed.

The speed of the rollout of HC platform throughout some entities needs to be improved, and in our research we attempted to investigate this issue using several methods. The exploratory interviews that we conducted were mostly centred on this client. Other organisations took part in the interviews that provided a valuable input to the design of the framework. We also conducted a survey for existing rollouts with the end users of this client.

The HC was also used for validation of the rollout framework that we designed. Both the findings of the exploratory interviews and the end-user survey are taken into account. The rollout framework was refined according to the evaluation of the HC management.

1.4. Research Questions

In this research, we aim to answer the following questions:

1. What can improve the speed of adoption for low-code/no-code applications?
   To address this question, we must first determine the factors that influence the pace with which low-code/no-code platforms are adopted.

2. Would offering a rollout method and template help?
   We will create a rollout framework after the factors have been determined. This framework will be evaluated to realise whether it can help speed up the rollout process.

3. Does the low-code/no-code platform technology affect the speed of adoption?
   The speed of adoption may be affected by several features of the low-code/no-code platform's technology and organisation. Our goal in this study is to comprehend the results.

1.5. Research Outline

The research is structured into the following chapters:

- **Chapter 2:** describes the methodology applied in this research using design science.
- **Chapter 3:** details the exploratory interviews that we conducted to identify the positive and negative factors that impacted the HC platform rollouts.
- **Chapter 4:** provides the end user survey details that we shared with users of the HC platform produced solutions. The survey utilizes the extended model of UTAUT. We measured the outcome of the degree of adoption of the HC platform technology to identify any underlying issues in the existing rollouts.

- **Chapter 5:** describes the design of the rollout framework using essential SAFe. The rollout framework is based on the feedback that we received using the exploratory interviews and the end user survey.
- **Chapter 6:** details the evaluation of the designed rollout framework that we performed with the HC management.
- **Chapter 7:** provides the refined rollout framework based on the evaluation outcome. The chapter covers two frameworks designed based on the need of the organisations.
- **Chapter 8:** covers the discussion and limitations of the research.
- **Chapter 9:** describes the conclusion, which includes the research contribution, the research questions that were answered, and potential future work.

2. Methodology

We used the design science methodology in our research. “Design science is the design and investigation of artifacts in context” [15]. In our research, the rollout framework is the artefact, and the context is low-code/no-code platform vendors, particularly the HC, and their clients who wish to smoothly rollout their solutions.

Using the Wieringa template [15], the design problem statement for our research is formulated as follows:

- Improve the adoption of the low-code/no-code platforms
- by designing a rollout framework
- that improves the speed of the rollout and overcomes the barriers
- in order for organisations to realise the benefits of using the platform at an early stage.

Peffers et al. built a process model of the design science research methodology (DSRM) that incorporated findings from several research papers and presentations. The model consists of six activities in a nominal sequence illustrated in below figure [7].

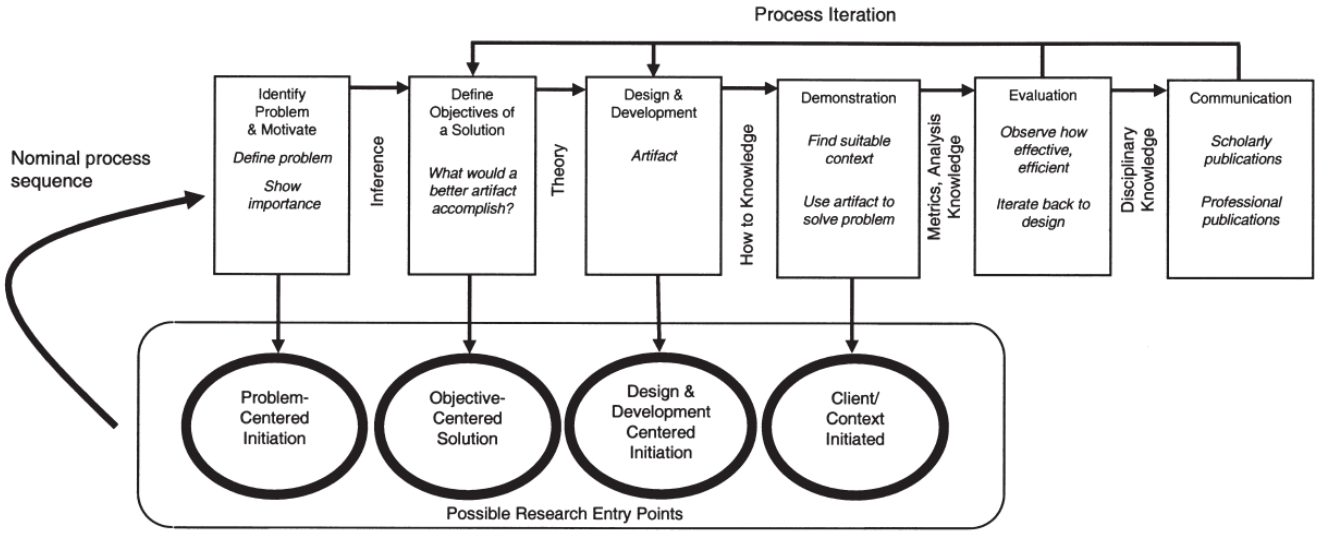


Figure 3 DSRM process model [7]

Below we discuss per activity how we performed it in our study.

Activity 1: Problem identification and motivation

The host company platform's rollout speed in some entities needs to be improved, and various factors and challenges were impacting the speed. As a result, the project's benefits were not realized as quickly as they should have been.

We used a variety of methods for the data collecting phase:

1. To understand the challenges and success factors of a new technology adoption, we examined several research publications as a foundation.

   Although the aforementioned research studies aid in the identification of various barriers and success factors, they do not address low-code/no-code platforms. We used some of these findings in our research to uncover the elements that affect the rollout of low-code/no-code platforms, particularly the HC.

2. We received training and access to platform of the HC. This gave us the opportunity to test it out and have a better knowledge of the platform before collecting further information.

3. We employed the qualitative method to gain a deeper understanding of the problems with the HC platform rollouts. To acquire a thorough grasp of the factors impacting the rollout plans, we gathered that information through exploratory interviews (see Chapter 3). The interviews were concentrating on one of the HC's clients and a small number of other potential clients. The relevant questions were formulated using the study articles as a guide.

4. We created a survey that was based on the findings of the interviews and observations to apply the quantitative method. The variables that affect the adoption of the platform of the HC were measured and identified in the survey using the UTAUT2 constructs.

   Momani evaluated numerous theories of acceptance of new technology implementation in 2020, and the UTAUT model was found to be the best fit [14]. In our research, we employed the extension of the UTAUT model to see if there were any major concerns that users had with the newly implemented entities.

   The survey was distributed among the end users that use the developed solutions using the HC platform (see Chapter 4).

5. The management of HC was aware of the data gathering in order to get their comments on the topics mentioned above and obtain further clarifications.

6. We also held other meetings with the CTO and architecture of HC to discuss the points that were raised and are relevant to the platform.

Once we finalized collecting the data, we were able to have a better plan for the design of the rollout framework.

Activity 2: Define the objectives for a solution

The objective is to design a rollout framework for low-code/no-code platforms that organisations can adopt in their digital transformation journey. The framework should consider all challenges that are identified in this research and make use of the factors that impact the rollout in a positive manner.

Activity 3: Design and development

The artifact for this research is the rollout framework. We designed the rollout framework that was built on data that we gathered in a variety of methods mentioned under activity 1. It considers essential SAFe within the different stages of the rollout (see Chapter 5).

Activity 4: Demonstration

After designing the rollout framework, we presented the tentative design of the framework to HC management using a PowerPoint presentation. The presentation was done with each participant separately. Further details about the presentation can be found in Appendix D.

Activity 5: Evaluation

We conducted the evaluation with HC management. A template based on the approach described by Riemenschneider et al. [23] was used to collect the feedback (see Chapter 6). Clients of HC were excluded from this evaluation due to the time constraints of the research.

Based on the feedback received, we modified the previous SAFe-based framework and created a new, more simplified framework (see Chapter 7). The design, development, and demonstration operations for this process have been repeated.

Activity 6. Communication

The master thesis paper documents all the research efforts and contributions, which we shared with the appropriate parties. Emails were used to distribute the thesis, which will also be posted on the Leiden University website.

According to Collatto et al [6], the essential elements for an adequate conduction of design science research are: Problem, solution, development, evaluation, adding value and publicizing. The elements listed in below figure were covered in the activities that we explained above.

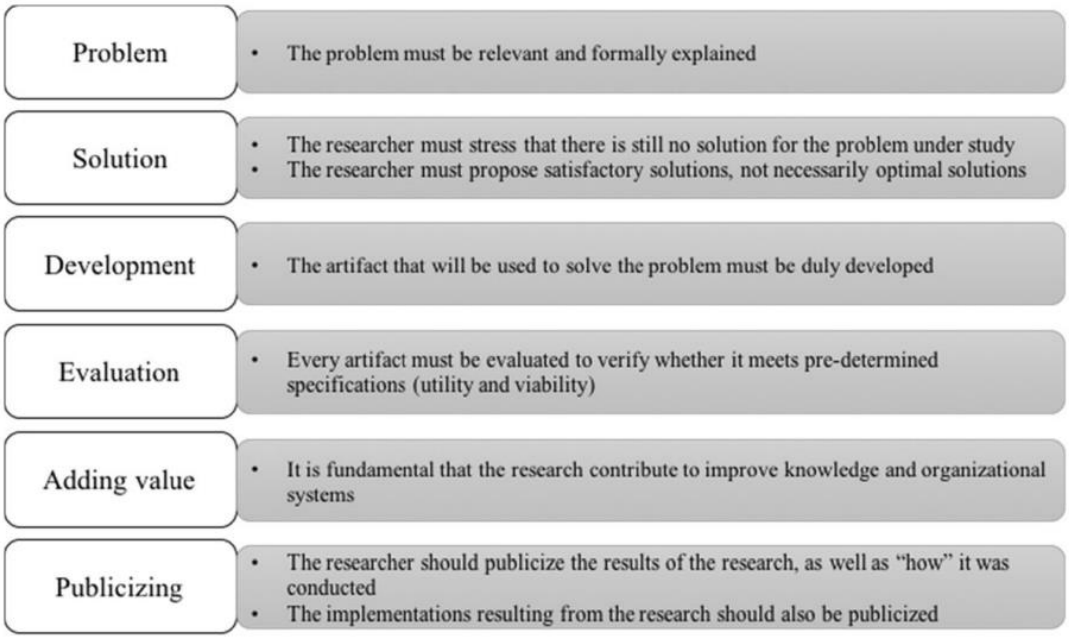


Figure 4 Elements conducting design science research. Source: prepared by the authors March and Storey (2008) [6]

Hevner believes that the design science must include the three research cycles: Relevance Cycle, Rigor Cycle and Design cycle, as shown in the figure below [5].

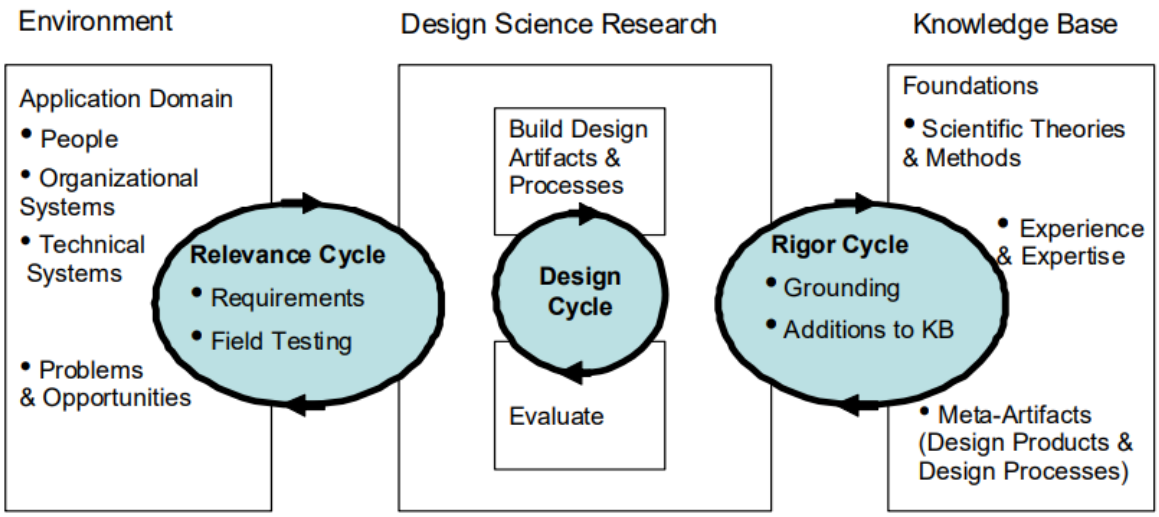


Figure 5 Design Science Research Cycles [5]

1. The Relevance Cycle: connects the activities of design science with the context of the research project. It covers the requirements and acceptance criteria. In our research, the requirements were established based on the interviews, survey, and using the HC platform. Additional iteration of the relevance cycle was not required following the evaluation of the rollout framework.

2. The Rigor Cycle establishes a link between the design science activities and the knowledge from scientific theories or methods that underpins the research project. We used scientific publications as a foundation of this cycle, however those studies did not address the problem meant to be solved from our research.

3. The design cycle iterates between the fundamental tasks of building and evaluating the designed artifacts. In our research, we conducted two design cycles based on the output of the evaluation for the designed rollout framework. The first cycle was evaluated by HC management and the second cycle was evaluated by HC CEO.

3. Exploratory Interviews

This chapter gets into the specifics of the exploratory interviews that we conducted for this study, as well as the findings and principles that will be used within the framework design. The goal of the exploratory interviews is to learn more about the challenges, success factors, and concerns that need to be addressed while rolling out low-code/no-code platforms.

Representatives from numerous departments, including Innovation, IT, Operations, and Procurement, participated in 14 interviews. Individuals from different levels of the focus company were interviewed to ensure that the issue was represented as broadly as possible. Among the interviews were employees from the headquarters, local terminals in various countries, and three responders from organisations that chose not to select HC technology. The interviews lasted an average of 40 minutes. The saturation point was reached in the tenth interview, when no new information was collected.

3.1. Sample Demographics

We conducted the exploratory interviews with the focus company along with other companies that did not proceed with selecting HC. Convenience sampling was used by contacting multiple interviewees by email to check their availability and willingness to participate. Anyone who played a role in the HC selection or HC rollout was included in our list of interviewees.

We proceeded with 14 participants who expressed an interest in taking part in the research. Large number of highly-experienced interviewees were included in the interviews. Because face-to-face interviews were not possible due to COVID-19 regulations, the interviews were done separately through Google Meet, Zoom or Microsoft Teams video calls. One participant used voice calls as a preferred communication method.

The demographic characteristics of the sample are summarized in below table:

Table 1 Descriptive information of the exploratory interview sample

Variable	**Value**	**Count**	**Percentage**
Gender	Male	12	86%
	Female	2	14%
Age	18-30	1	7%
	31-40	6	43%
	Above 40	7	50%
Experience (Total in Years)	0-10	4	29%
	Above 10	10	71%
Organisation / Location	Focus Company Head Quarter	7	50%
	Focus Company Local Terminal	4	29%
	Did not proceed with HC	3	21%
Total		14	

3.2. Literature review for the Exploratory Interview

We used three research articles as a foundation for the exploratory interviews. The articles contained some useful material that we used as a starting point for our exploratory interviews. Some of the content in the articles didn't apply to the low-code/no-code platforms or the interviews we performed, so they weren't used in the research..

The first source addressed the drivers, success factors, and barriers to digital transformation in the maritime transport sector. Where relevant, we incorporated parts of those drivers, success factors, and barriers in the exploratory interviews. The research analysed 95 sources to come up with the results. Success factors are aspects essential for attaining the intended goals, while Barriers are significant obstacles to the digital transformation process. Drivers are external or internal causes that motivate organizations to participate in digital transformation [9].

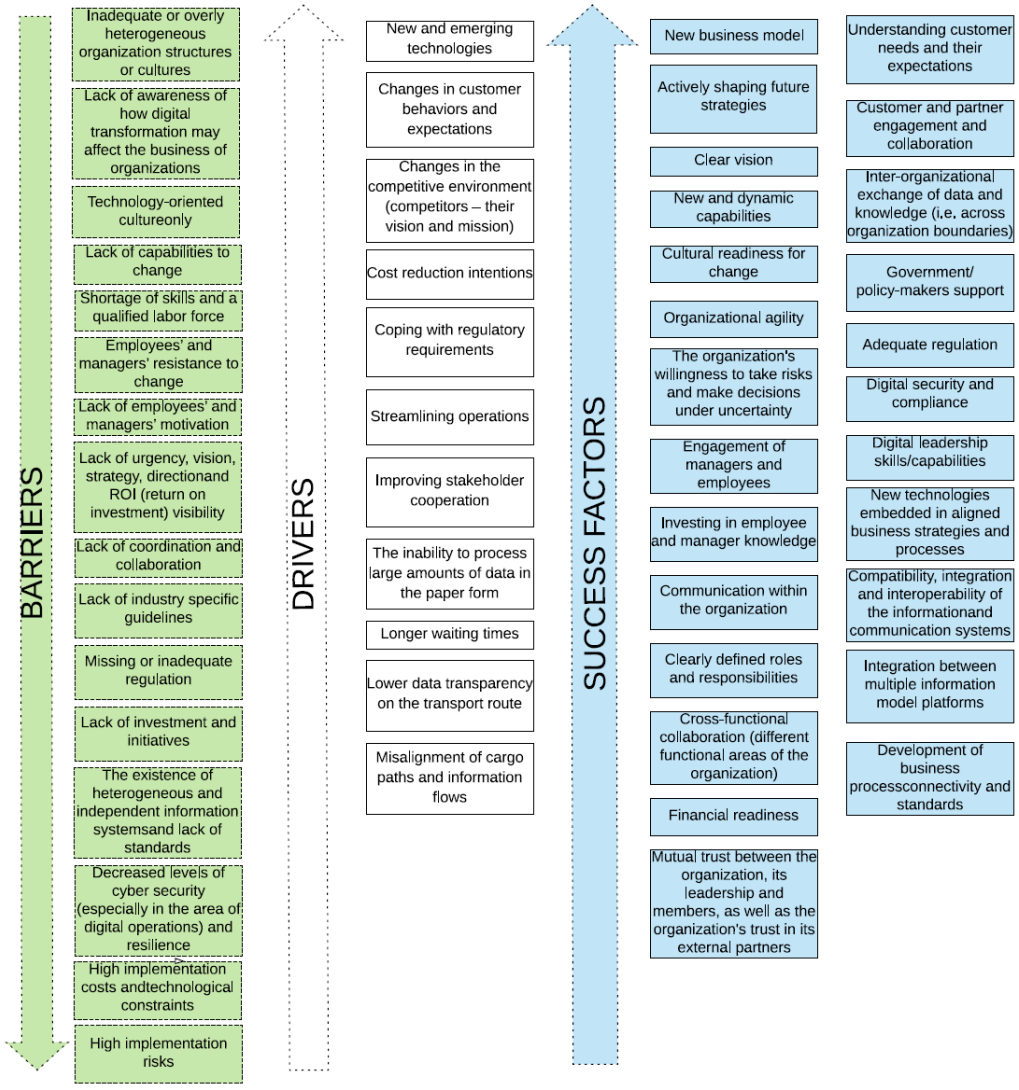


Figure 6 Model of drivers, success factors and barriers affecting digital transformation in the maritime transport sector [9]

The study paper's identified drivers, barriers, and success factors that were categorised into organisational, technological, and external environment contexts. We took these contexts into account during the exploratory interviews.

The second source considers the change implementation model with the phases: Vision Phase, Resource Commitment Phase, Design Phase, Alignment Phase and the Action Phase with their Exit criteria [10]. The obstacles and success factors are summarised in below table:

Table 2 Implementation obstacles and success factors [10]

Process Implementation Obstacles	**Key factors for success**
1. The Lack of Vision Problem 2. IT Alone is Not Enough 3. The Mis-alignment with People Problem 4. The Burden of IT Legacy Systems 5. IT Implementation Problems	1. Clear Vision and Focus on Results 2. Alignment with Organisation's Environment 3. Infrastructure Support 4. The Importance of User Input

According to Yeh [10], one of the major obstacles include the lack of vision problem since some technological solutions only address a portion of the whole problem. A key element in overcoming this is the managerial vision and focus on the business need. IT introduction alone is insufficient; further organisational and role-level modifications are needed. The human resource management should handle this and make it obvious to the resources that using IT may result in downsizing or demand more capabilities from them.

Another factor to take into account is IT legacy systems. Management should determine whether to build internally from scratch or buy a platform to support application development. Legacy systems, data migration, the location of the rollout, the infrastructure, resource costs, or other issues may create implementation challenges. The approach should be flexible, allowing for the usage of prototypes, trial rollouts, and user input throughout the process.

We considered some of the obstacles and success factors above that were relevant to the situation during the exploratory interviews.

The third source addresses the barriers to successful implementation of reengineering and identifies seven critical barriers for its success [11]:

1. Misunderstanding of the concept
2. Misapplication of the term
3. Lack of proper strategy
4. Unrealistic objectives
5. Management failure to change
6. Failing to recognize the importance of people
7. IS failure to change

We analysed the information from the three sources, and the points that were significant to low-code/no-code platform rollouts were grouped into nine categories. The nine categories include: vision and strategy, social influence, finance, resources, stakeholder management, technology, data management, culture and regulatory. Whenever one of those categories was addressed during the interviews, the points associated with it were assessed based on the circumstances.

3.3. Exploratory Interview Process

We started each interview with a summary of our research and its purpose. The next action was to learn more about the interviewee's background and goals for using HC. During the

interviews, we utilized a form to fill in the responses, which also allowed for the possibility of further issues being highlighted. Further details about the form can be found in Appendix A.

Background Information	
HC Implementation	
Vision and Strategy	
Social Influence	
Finance	
Resources	
Stakeholder Management	
Technology	
Data Management	
Culture	
Regulatory	
Global vs Local issues	
Other points raised	

Figure 7 Exploratory interview form

3.4. Exploratory Interviews Findings

Interviewees from the focus company with local terminal and headquarters locations were considered in the analysis of the interviews. Other participants that did not progress with HC selection were eliminated since they could not offer input on the rollout challenges and success factors. Their feedback, on the other hand, was valuable and was taken into account in the design of the rollout framework within the problem definition and technology selection stages.

The findings of the exploratory interviews were classified into organisation, technology and external environment. In our research, we applied the same contexts as Tijan et al. did [9]. When a topic came up during the interviews, we would tag it and place it in one of the three relevant contexts. Based on the established tags, categories arose from our analysis of the interview data and were grouped by topic.

The table below summarizes the various categories found in each of the three contexts.

Table 3 Exploratory interviews findings contexts

Organisation	**Technology**	**External Environment**
Management Support	Data	HC Aspects
Roles and Responsibilities	Cost	Regulations
Change Management	Architecture and Security	
Customer Journey and Experience		
Budget Allocation		
Human Resources		

The finding contexts are summarized and explained in depth in the following sections:

Organisation

The organisation context covers the categories relevant to the internal aspects of the organisation that implements and rolls out the low-code/no-code platform. The categories include:

- Management Support: refers to the influence of the management in supporting the rollout and the usage of the low-code/no-code software.
- Roles and Responsibilities: refers to the clear roles and responsibilities required for a rollout in all stages.
- Change Management: refers to the process of implementing the change. This includes the methodology followed and the decision on what to standardise or customize based on the local terminal needs.
- Customer Journey and Experience: refers to having one experience in all terminals for the same process.
- Budget Allocation: refers to the budget allocation setup between headquarters and local terminals including the invoicing setup. The budget covers the licensing, customisations, training and devices.
- Human Resources: refers to different aspects of the human resources. It includes the skills and knowledge required for the resources, availability of the resources, their motivation to use the platform and any organisational chart changes required for the low-code/no-code platform rollout.

Technology

The technology context covers the categories relevant to the technological development of the low-code/no-code platform and any other integrations involved in the process. The categories include:

- Data: refers to the low-code/no-code platform's data collection and management.
- Cost: refers to the TCO (Total Cost of Ownership) of the low-code/no-code platform in addition to the time required to have it in place.
- Architecture and Security: refers to the architecture design and the security aspects that are considered while scaling and integrating with other technologies.

External Environment

The external environment context covers the categories that are outside the organisation but affect it in some situations. The categories include:

- HC Aspects: refers to aspects related to the organisation of the low-code/no-code software such as support and performance in addition to the design of the solution provided.
- Regulations: refers to the regulations imposed by the government that affect the rollout such as some regulations related to the Covid-19 issue.

Below treemap charts show the contexts and their factors. They display the number of interviewees that mentioned this factor as a positive or negative impact on the rollout. Some factors can be found in both charts since they have an impact on the rollout in both directions. As example, consider the cost: some cost savings can be made, which accelerates the rollout, but there is also a high cost on the other hand, which may slow it down.

The treemap's colours represent the various contexts considered in the interviews. The technological context is represented by yellow, the organisation context is represented by blue, and the external environment context is represented by green. Each rectangle's size corresponds to the number of interviewees who reported the factor. The larger the rectangle size, the more often this factor was noted by interviewees.

The findings for the positive impact indicates prime factors impacting the speed of the low-code/no-code platform rollout. Those factors include: digitizing the process and gathering data, the management support for using HC and cost reduction.

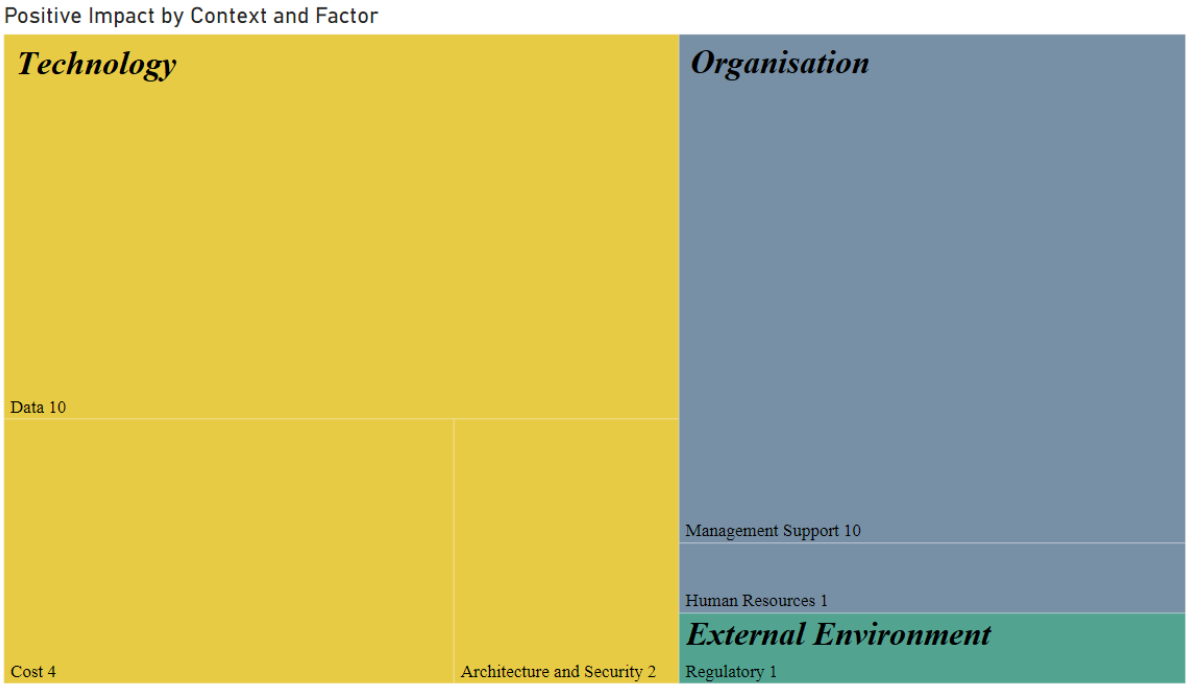


Figure 8 Positive impact by context and factor

On the other hand, different factors hinder the adoption of HC and impact it negatively Those factors include the unclear budget allocation between the headquarters and local terminals, unclear roles and responsibilities, incomplete change management setup, human resources challenges, and incomplete architecture design.

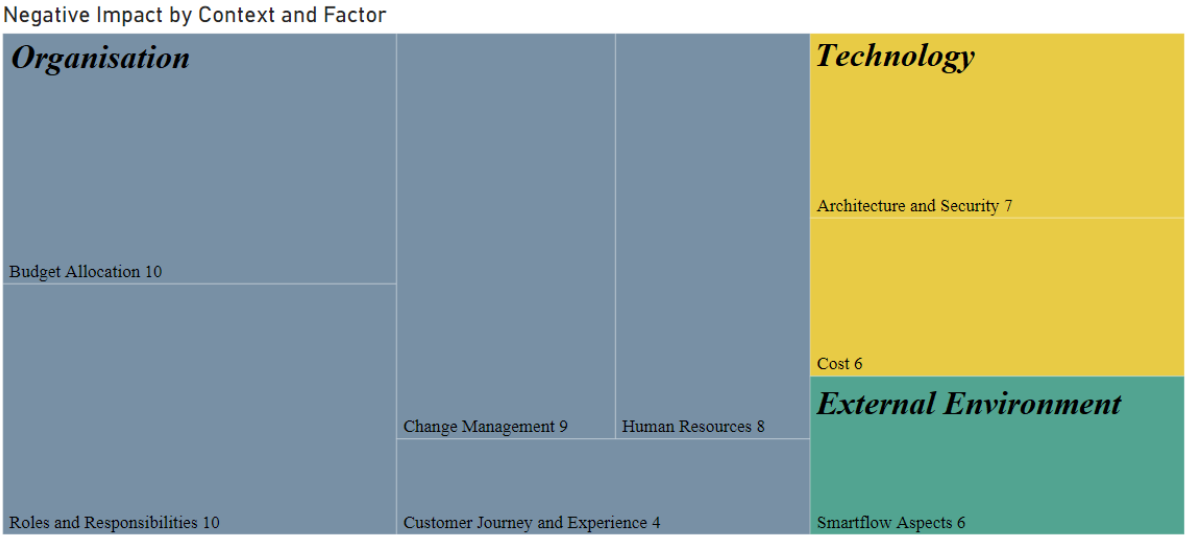
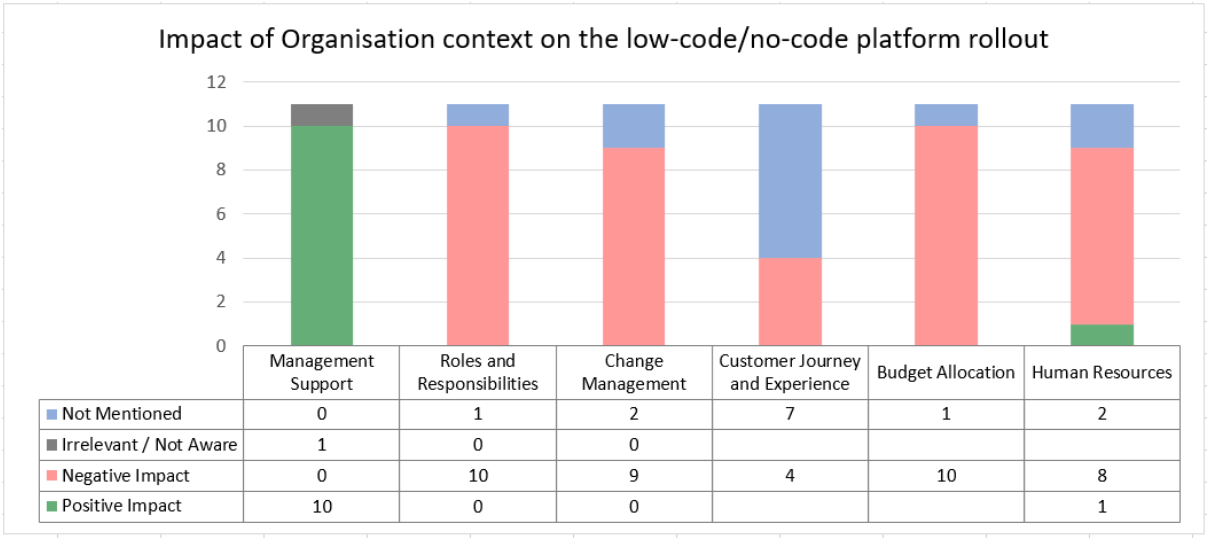
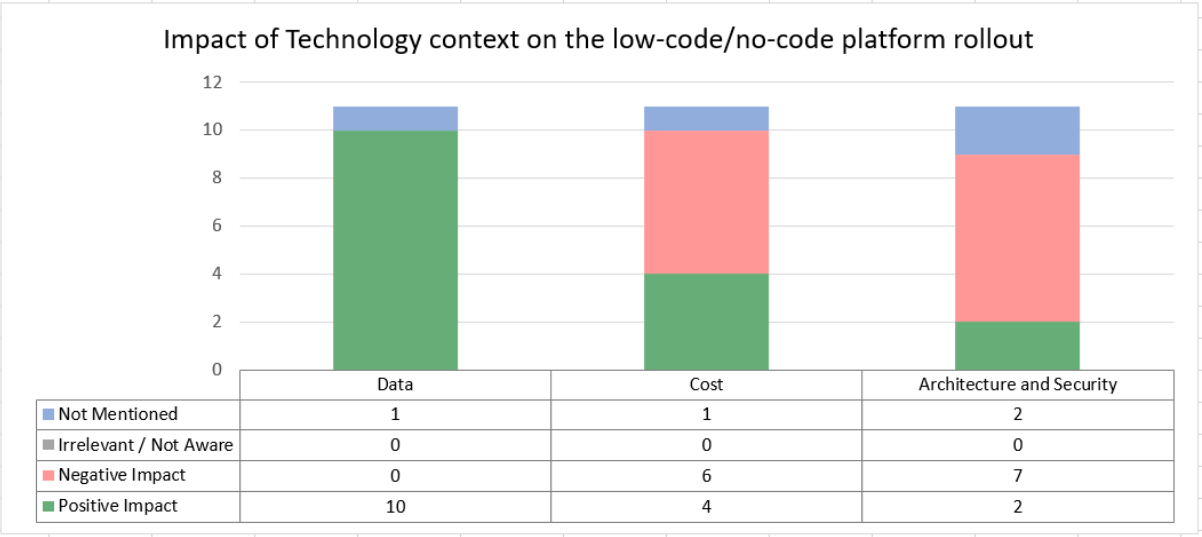


Figure 9 Negative impact by context and factor

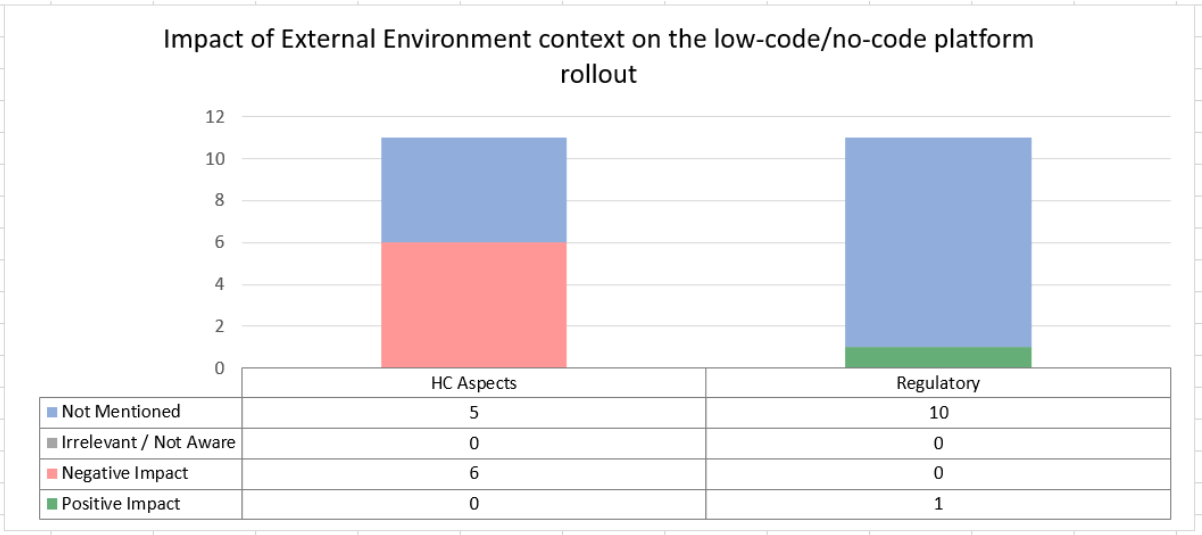
The impact of the factors is reflected in the bar charts below, which show the number of respondents who described the factors as positive (in green), negative(in red), neutral(in grey), or did not mention them at all(in blue). The charts display each context separately:



	Management Support	Roles and Responsibilities	Change Management	Customer Journey and Experience	Budget Allocation	Human Resources
Not Mentioned	0	1	2	7	1	2
Irrelevant / Not Aware	1	0	0			
Negative Impact	0	10	9	4	10	8
Positive Impact	10	0	0			1



	Data	Cost	Architecture and Security
Not Mentioned	1	1	2
Irrelevant / Not Aware	0	0	0
Negative Impact	0	6	7
Positive Impact	10	4	2



	HC Aspects	Regulatory
Not Mentioned	5	10
Irrelevant / Not Aware	0	0
Negative Impact	6	0
Positive Impact	0	1

Figure 10 Impact of the contexts on the low-code/no-code platform rollout

Each context has both positive and negative impacts. Based on the interview content, the following are the specifics of the impact shown in the above charts:

Organisation Context Impacts

Table 4 Positive and negative impact on the rollout – Organisation context

	Positive Impact	**Negative Impact**
Management Support	• Management supports the rollout of the software • Pushed by the innovation team	
Roles and Responsibilities		• Unclear roles and responsibilities • The decision to manage it globally or on regional or local level
Change Management		• Not decided yet what is standard vs what needs to be customized • Insufficient requirements gathering • Slowness of involving different departments • Manage the huge backlog for local requirements
Customer Journey and Experience		• Concern about having one experience across the terminals
Budget Allocation		• Budget allocation setup between the local terminals and headquarters is unclear • Invoicing setup per terminal • Limited local terminal budget
Human Resources	• Motivated users encourage the usage of the platform • Resources knowledge in using smartphones • Local knowledge in developing apps using the platform	• Training required to have the needed skills • Time invested for knowledge transfer • Bottlenecks by the headquarters to provide support for local terminals • Resources unavailability due to holidays or workloads • Organisational chart changes to setup a team for managing the platform • Extra roles required such as Project managers, builders, and post implementation support

Technology Context Impacts

Table 5 Positive and negative impact on the rollout – Technology context

	Positive Impact	**Negative Impact**
Data	• Future plans for reporting and dashboards • Make use of RPA and APIs • Solved the handwriting issues • Better management of the data in bad weather conditions • Less human errors • Faster data entry • Clear steps and statuses of the process • Up to date Data • Data transparency for users and management • Reusing existing data • Integrations with other systems • Extra safety achieved	
Cost	• Benefits are higher than cost if properly rolled out • Save costs of hiring extra resources, resources can focus on other tasks • Lower outsourcing costs • Lower the cost of mistakes	• High training budget • Extra cost for customizations • High TCO (Total Cost of Ownership)
Architecture and Security	• Easy integration with other systems	• Complex architecture design for scaling in a big organisation • Security policies in place to be considered when rolling out • Rigid devices causing slower connections • Hardware availability issues

External Environment Context Impacts

Table 6 Positive and negative impact on the rollout – External environment context

	Positive Impact	Negative Impact
HC Aspects		• Performance issues • Dashboard limitations • 24hrs support unavailability • Redesign after POC
Regulatory	• Covid-19 related governance regulations to push for remote work	

3.5. Factors with positive impact on the rollout speed

During the exploratory interviews, we identified positive impacts of different contexts on the rollout speed. According to the interviewees, the positive impacts that speed up the rollout of the low-code/no-code platforms under the three contexts include:

- **Management support:** It is important to have the management support when rolling out the low-code/no-code platform. As a result the approval from the management should be obtained in different stages.
- **Data management:** Using low-code/no-code platforms to capture data is an important aspect of the implementation. It aids in the integration of other systems as well as the reporting of up-to-date data. Another essential matter is lowering inaccuracy rates and the reusability of the data which has an impact on costs and safety precautions.
- **Cost:** Proper deployment of low-code/no-code platforms reduces costs by decreasing inefficiencies and the need for employing or outsourcing resources for the manual work.
- **Architecture and security:** Scalability is achieved in deploying low-code/no-code platforms in the organisation with easy integrations with other systems.
- **Regulatory:** COVID-19 legislation in some countries, like as Singapore, pushed the usage of low-code/no-code platforms by mandating remote solutions in a short period of time.
- **Human Resources:** Where possible, motivated users urge other users and departments to use low-code/no-code platforms in their operations. The end-user may easily use the produced solutions considering the low platforms' simple user interface. It also enables users to quickly construct applications utilizing the platforms without requiring substantial IT skills.

3.6. Factors with negative impact on the rollout speed

We identified that diverse circumstances had detrimental effects on the rollout speed. The following factors slow down the rollout of the low-code/no-code platform:

- **Roles and responsibilities:** The rollout is slowed by unclear roles and responsibilities. Various discussions will take place in order to determine who is accountable for making decisions or approving modifications.

- **Budget Allocation:** At the start of the rollouts, the budget distribution between different entities and the headquarters should be properly set up. Otherwise, the ambiguity will hinder implementations, since entities may have limited budgets and are unwilling to pay for new implementations. To ensure a seamless deployment, the invoicing setup with the low-code/no-code organisation should be agreed upon.
- **Change management:** Part of the slowness is due to the gathering and approval of requirements, as well as prioritizing and stakeholder management. To solve these issues, the organisations need a documented procedure in place.
- **Human Resources:** Training, time spent on knowledge transfer, as well as support setup and resource availability, are all necessary to keep rollouts on track.
- **Architecture and security:** In certain circumstances, a complicated implementation necessitates a thorough discussion of the architecture design and security concerns in order to minimize reworks and delays.
- **Cost:** While low-code/no-code platforms save expenses, they may also have a high total cost of ownership (TCO), which includes any further customizations and training that may be required. In some cases, entities might not be willing to pay this extra cost.
- **Low-code/no-code platform aspects:** The availability of support and the platform's performance are major factors that influence rollouts. Some platform restrictions may also contribute to the slow adoption of a platform.
- **Customer journey and experience:** It's critical to provide the same experience across multiple entities when rolling out platforms to end users. There may be some complications in some entities due to local requirements.

3.7. Theory of rollout speed for low-code/no-code platforms

In this section, we will construct the theory based on the exploratory interview findings.

From the product selection through the implementation of the solution, the process of rolling out a low-code/no-code platform necessitates careful consideration of a number of criteria. Low-code/no-code platforms are evaluated through a vendor selection process that takes into account a variety of factors. The User Interface, connections with other systems, data management, administration, scalability, and other critical characteristics of the solution are only a few of the components. Each factor has a distinct weight and, if not accounted for in the early stages of the project, might be a show-stopper or a barrier in the future.

It is vital to have the management support once the product has been chosen in order to ensure a successful rollout. One of the interviewees in the exploratory interviews stated that the lack of management support is impacting the implementation on a bigger scale. Interviewees from other organizations, on the other hand, indicated that the presence of management support facilitates the decision to proceed with the chosen solution.

Following the establishment of management support, the next stage is to determine the change management approach that will be used to deploy the solution. This procedure should lay out the processes in detail, as well as the roles and responsibilities that will be necessary to oversee the project's execution. To guarantee a seamless deployment, the budget allocation and architecture must also be agreed upon. The architecture design must take into account the demands of the corporate headquarters and local entities to enable future scalability.

During the final step of installing the solution in each local entity, a training plan should be designed to ensure that proper assistance is provided and knowledge is transferred to the appropriate resources. In order to avoid affecting the go-live dates, it is critical to consider resource availability throughout the planning process.

Government restrictions may have a role in accelerating a deployment in some instances. Singapore has implemented rules to reduce the pace of Covid-19 transmission where the organisations' employees had certain criteria for being on-site. This necessitated the development of alternatives to the paperwork used in the inspections, as well as a faster solution using tablets and low-code/no-code platforms.

We did not detect any major concerns after the applications are handed out to end-users. Personnel in the inspection business are familiar with mobile applications and, in some circumstances, utilize other applications in their everyday job. In this case, the shift from paper to tablet is seamless.

We will design the framework using the principles outlined above. They will be taken into account in order to improve the chances of a faster rollout.

4. UTAUT Model for existing Rollouts

We utilized a survey to identify whether there were any problems with the current HC implementations. The objective is to evaluate HC's implementation and adoption by the focus company's end users. We sent the survey to end users in order to identify any insights that can then be integrated into the rollout framework's design. It was widely distributed to people in several regions who use HC for different processes. The survey was shared using an email containing a link to Qualtrics survey software. Further information about the survey is under (Appendix B).

The Unified Theory of Acceptance and Use of Technology (UTAUT) model was used in the survey to measure the technology acceptance. “The UTAUT is a useful tool for managers needing to assess the likelihood of success for new technology introductions and helps them understand the drivers of acceptance in order to proactively design interventions (including training, marketing, etc.) targeted at populations of users that may be less inclined to adopt and use new systems” [12].

Eight different models were identified to create the conceptual framework of the UTAUT model: Theory of Reason Action, Theory of Planned Behavior, Technology Acceptance Model (TAM), Motivational Model, Combine Theory of Planned Behavior and Technology Acceptance Model, Model of PC Utilization (MPCU), Innovation Diffusion Theory and Social Cognitive Theory. UTAUT will give a solid foundation for describing why users accept or reject a technology in a particular context [13].

According to a comparison performed by Momani between UTAUT and previous models, there are two types of technology acceptance theories. First type is the restricted theories that can’t be deemed as comprehensive due to their limited constructs such as TAM. Second type is the comprehensive theories that include a lot of constructs and are hard to implement such as MPCU [14]. Momani observed that while the UTAUT is the most thorough model for examining technology's real usage behavior, it is not very generic. The model has limited number of constructs and moderating variables built with the least level of complexity. This makes it the feasible theory to investigate the acceptance of implementing new technologies.

UTAUT2; which is an extension of the UTAUT model, incorporates three additional constructs into UTAUT. The constructs are: hedonic motivation, price value, and habit [16]. We measured the hedonic motivation and habit in the end user survey. Price value in UTAUT2 was excluded since the survey was sent to end users who use the application and are unaware of the associated expenses. All constructs were measured using Likert Scale questions ranging from 1 to 5, with 1 indicating strong disagreement, 3 neutral, and 5 indicating strong agreement.

While we make use of the UTAUT model to construct our end user survey, our objective was not to statistically test the hypotheses that are embedded in the model, but to simply obtain any interesting insights that may inform our design of a rollout framework. For the former objective, a large number of respondents would have been necessary, but for the latter purpose a smaller number of respondents can already bring value.

According to the findings the variables gender, age, and experience do not moderate the influence of the constructs on the behavioural intention to use HC. The overall survey results show a strong intention to adopt using HC.

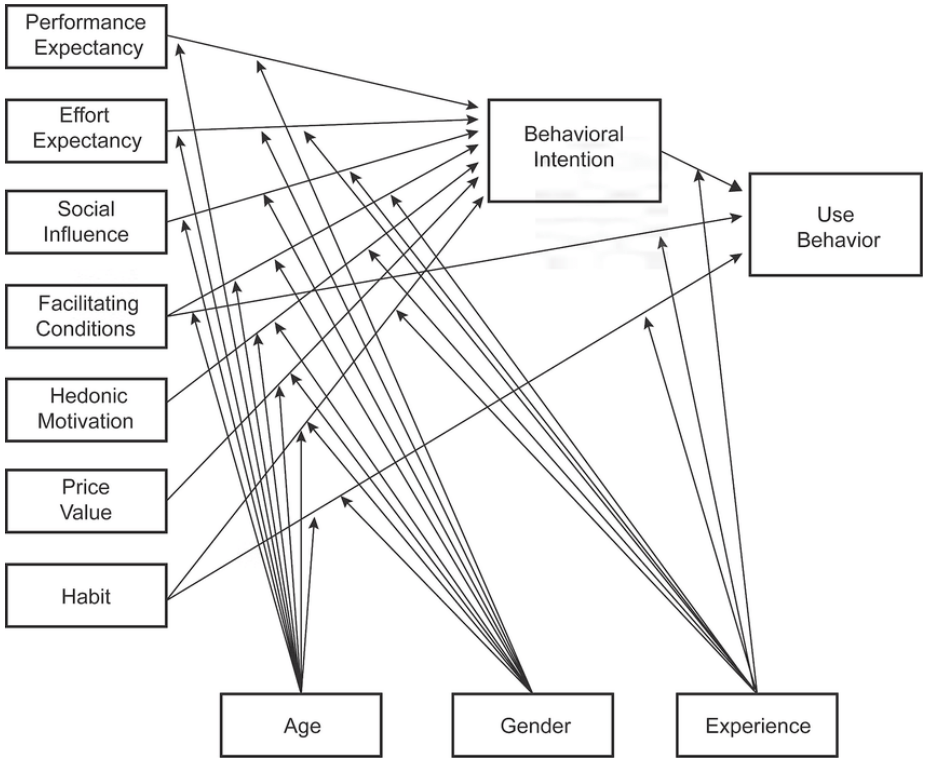


Figure 11 Unified Theory of Acceptance and Use of Technology (UTAUT2) Model [16]

4.1. Survey Considerations

A total of 53 people responded to the survey, which we distributed to users who use HC developed solutions as part of their profession. We eliminated four responses because they stated that they had never used HC. The remaining 49 responses were included in the research.

We divided the age and experience into groups based on the sample size. Any group with fewer than ten people was merged. Because the responses were imprecise and might lead to inaccurate interpretation, "others" was omitted from reporting usage frequency. Due to the minimal number of females in the gender findings, the responses were not analysed in terms of gender.

4.2. Survey Demographics

The demographic characteristics of the users are summarized in below table:

Table 7 Descriptive information of the survey sample

Variable		**Count**	**Percentage**
Gender	Male Female	43 6	88% 12%
Age	<35 35-45 Over 45	16 13 20	33% 27% 40%
Experience (Total in Years)	0-9 Above 10	12 37	24% 76%
Highest Education	High School University or College	27 22	55% 45%
Total		**49**	

4.3. UTAUT Model Moderating Variables

The following variables were measured in the survey sent to the end users:

- Age: The age of the end user.
- Gender: Female, Male or Others.
- Experience: The total work experience in years in addition to the highest education degree.

The age of the user is a critical factor in ensuring that experienced users have no difficulties using the platform. On the other hand, neither education nor total work experience had a significant influence on the behavioural intention to use HC or the constructs. We didn't include the gender variable in the reporting since the gender finding has a small number of females, making it difficult to analyse.

4.4. Performance Expectancy Construct

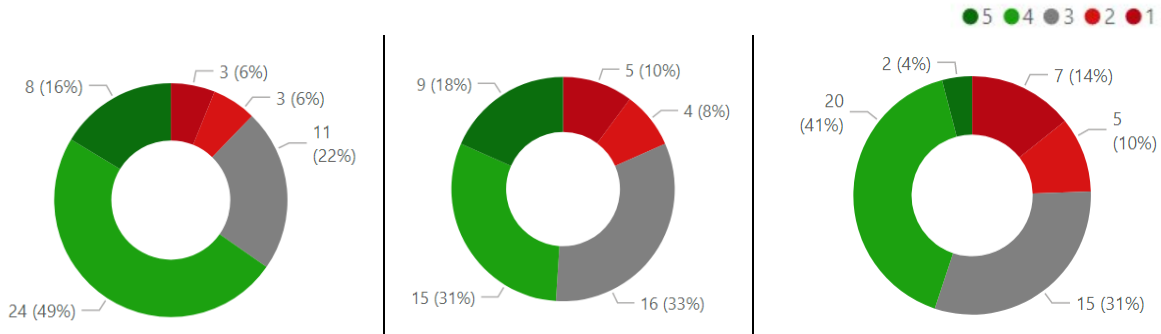
Performance Expectancy is the degree to which an individual feels that adopting the system will assist him or her in improving work performance [12].

This construct is measured by the improvements achieved, the speed of tasks and the low-code/no-code platform performance after implementing the technology using the following questions:

- Q1: I feel improvements are made in my work after using HC
- Q2: HC helps accomplish things faster
- Q3: I'm satisfied with HC performance (speed)

In this construct, the overall responses suggest a high level of satisfaction. Experienced users over the age of 45 can also see improvements in their job. High-frequency users who utilize the platform on a daily or hourly basis provide positive feedback. There is, however, opportunity for HC performance improvements in some use cases, which were under development while performing the survey and were deployed during Q1 2022.

Some replies do not imply favourable feedback, as the goal of using HC in some use cases is to digitize rather than speed up existing processes. This was brought up during the exploratory interviews by some interviewees when the goal of introducing HC was addressed.



Q1: Improvements are made | Q2: Accomplish tasks faster | Q3: HC Performance

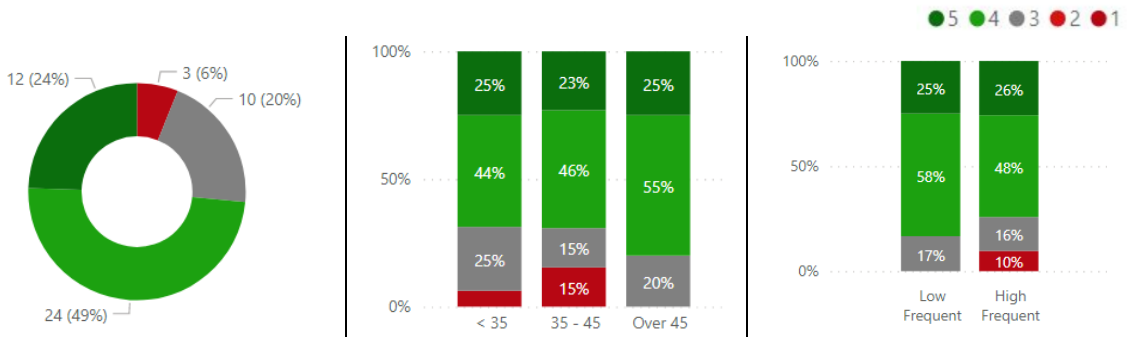
Figure 12 Responses for performance expectancy construct per question

4.5. Effort Expectancy Construct

Effort Expectancy is the degree of ease with which the system may be used [12]. This construct is measured using the following question:

- Learning how to use HC is easy for me

The overall satisfaction with the ease of learning how to use HC is fairly high, particularly among those over the age of 45.



Overall Responses | By Age Group | By Usage Frequency

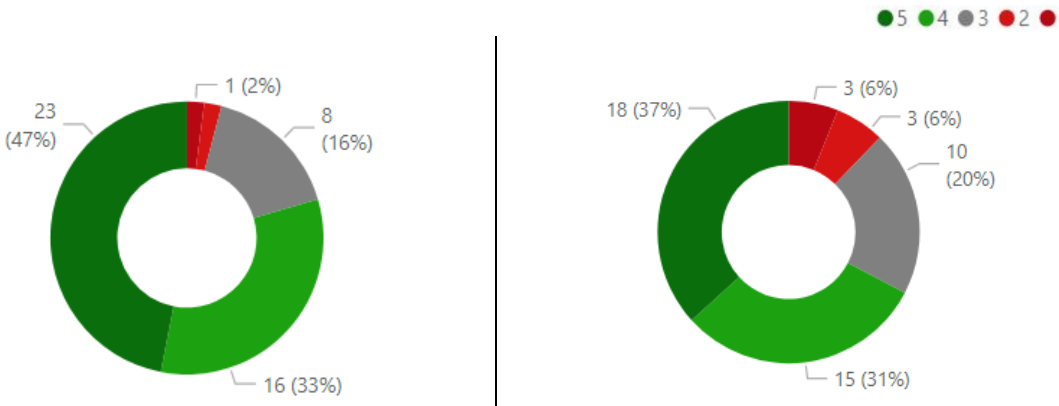
Figure 13 Responses for effort expectancy construct

4.6. Social Influence Construct

Social Influence is the degree to which an individual believes that the acceptance of utilizing the technology is influenced by others around them. This construct is measured using the following questions:

- Q1: The management thinks that I should use HC
- Q2: HC is useful for my co-workers

Overall, the results demonstrate that HC is beneficial in different use cases and it has a high degree of management support.



Q1: Management Support | Q2: Usefulness for Co-workers

Figure 14 Responses for social influence construct per question

4.7. Facilitating Conditions Construct

Facilitating Conditions is the degree to which an individual believes that the knowledge required to use the technology exists using the organisational and technical infrastructure. This construct is measured using the following question:

- I have the knowledge necessary to use HC

Although the majority of respondents stated they have the essential knowledge to use HC, a tiny minority (around 10%) claimed they didn't. This group will require further training to guarantee that the information is available for them.

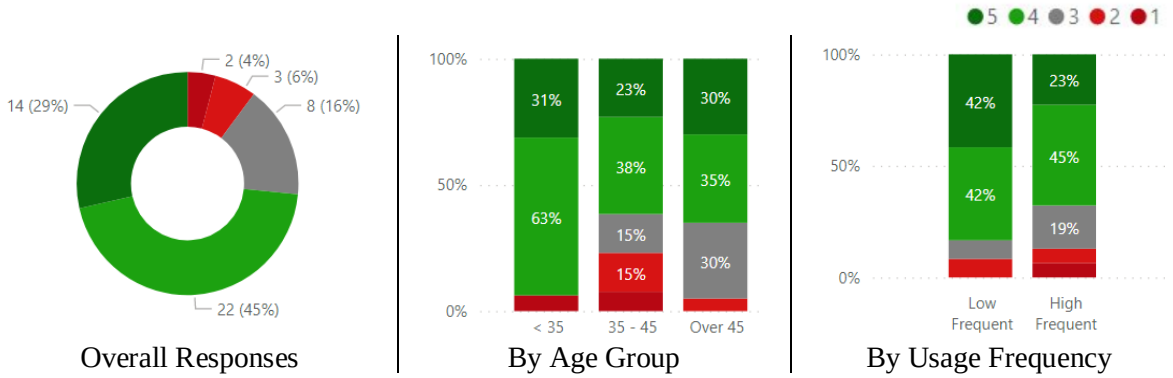


Figure 15 Responses for facilitating conditions construct

4.8. Hedonic Motivations Construct

Hedonic Motivations is the associated pleasure experienced in using a technology. This construct is measured using the following question:

- Using HC is enjoyable

The goal of implementing HC is to digitize a process, but it's evident that users of all ages and frequency of use are enjoying the platform.

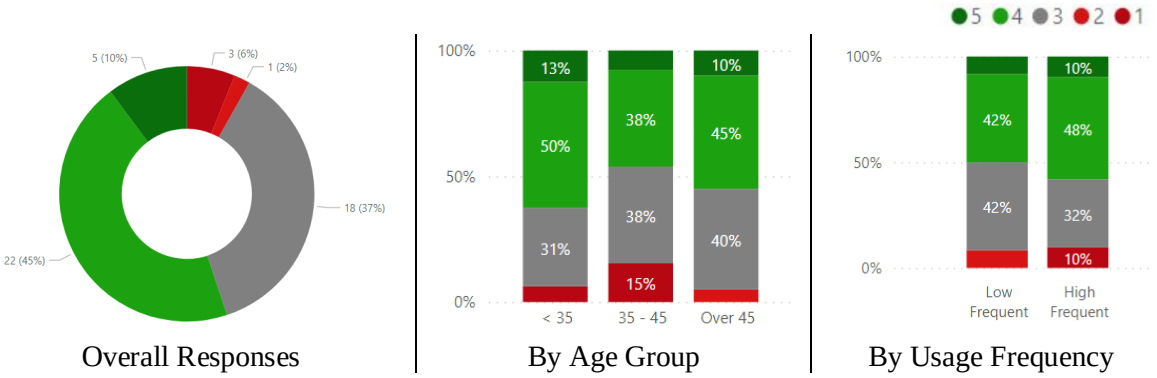


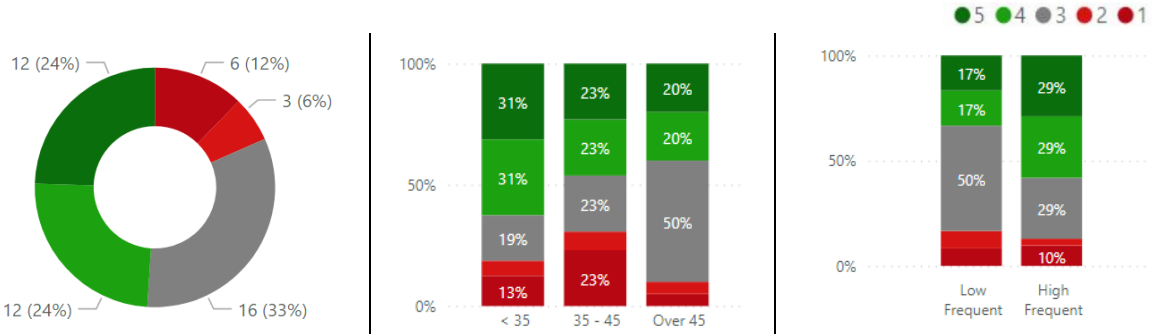
Figure 16 Responses for hedonic motivations construct

4.9. Habit Construct

Habit refers to the automatic behaviour using a technology. This construct is measured using the following question:

- Using HC became a habit

Different factors, such as whether the process was recently implemented, must be evaluated further, since this may have an influence on the findings. The findings are generally promising; those who reported high or very high habit formation are nearly twice as large for the high frequency usage compared to the low frequency usage. This demonstrates that HC use patterns are being created for the daily and hourly usages. On the other hand, as people become older, it is clear that habit formation is declining.



Overall Responses | By Age Group | By Usage Frequency

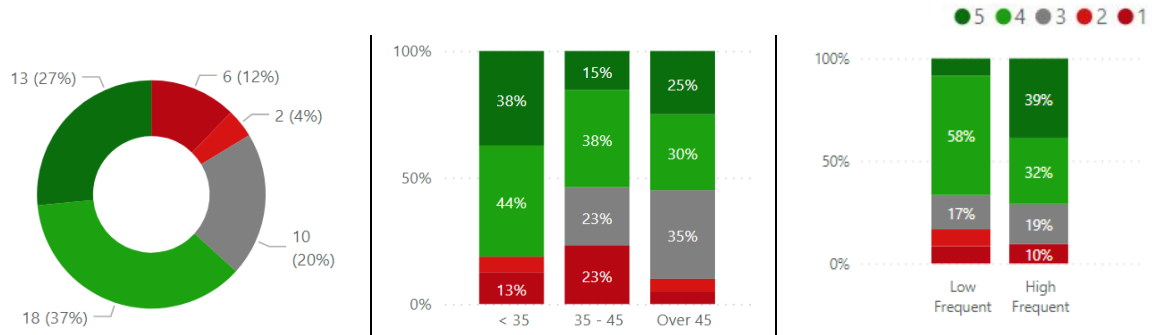
Figure 17 Responses for habit construct

4.10. Behavioural Intention

Behavioural Intention refers to the individuals' intention to use a particular technology. This is measured using the following question:

- I always try to use HC

A substantial percentage of respondents desire to utilize HC, which is a great outcome in terms of platform adoption. A minority of 16% of the respondents will need their departments to persuade them of the benefits of implementing HC in their professions.



Overall Responses | By Age Group | By Usage Frequency

Figure 18 Responses for behavioural intention

4.11. Survey Findings

The overall survey reveals a good intention to use HC for the entities that have been rolled out. Further information is available under (Appendix C).

Based on the Likert scale questions with values ranging from 1 to 5, the median and mean values for the UTAUT constructs are as follows:

Table 8 Median and mean for UTAUT constructs results

Construct	Median Value	Mean Value
Performance Expectancy		
• Improvements are made after using HC	4	3.6
• HC helps to accomplish things faster	3	3.4
• HC Performance	3	3.1
Effort Expectancy	4	3.9
Social Influence		
• Management Support to use HC	4	4.2
• HC usefulness for Co-Workers	4	3.9
Facilitating Conditions	4	3.9
Hedonic Motivations	4	3.5
Habit	3	3.4
Behavioural Intention	4	3.6

The key findings for the constructs that are essential for the rollouts are summarized as follows:

- **Performance Expectancy:** Overall, the responses indicate a high degree of satisfaction with the low-code/no-code platform's performance. We identified an improvement that can be achieved in the monitoring and support stage to show any performance issues for the low-code/no-code platform.
- **Effort Expectancy:** Overall, end users including the ones over the age of 45 are pleased with how simple it is to learn how to utilize HC.
- **Social Influence:** Management is in favour of implementing the low-code/no-code platform, and a large number of end users feel that HC will benefit their co-workers.
- **Facilitating Conditions:** Although the majority of end users answered that they had the necessary knowledge to utilize HC, about 10% respondents stated the opposite. We determined that training for this group should be considered in order to guarantee that they have the necessary conditions to perform.
- **Hedonic Motivations:** End users of all ages and frequency of use are enjoying the platform.
- **Habit:** HC use patterns are being created for the daily and hourly usages. The habit formation is declining for experienced users of older age. We determined that a proper feedback procedure will be required in order to understand the challenges they experience and take appropriate corrective action.
- **Behavioural Intention:** A substantial percentage of respondents desire to utilize HC, however 16% of the end users will require their management to persuade them of the benefits of implementing HC in their professions.

4.12. Considerations for the rollout framework

Based on the survey findings, we should consider the following while designing the rollout framework:

1. Training
   We examined the facilitating conditions and 10% of the respondents stated they did not have the knowledge necessary to use HC. To solve this problem, we need to consider adequate end-user training that takes into consideration all age groups.

2. Vision
   We measured the behavioural intention to use HC and found that 16% of the respondents did not want to use it all of the time. The framework should make sure that the low-code/no-code platform's vision is conveyed to the teams. We need to persuade end users of the advantages of rolling out HC with management's support.

3. Feedback
   Some survey responses necessitated additional explanations in order to determine which processes need to be improved and what knowledge is necessary to improve existing rollouts. Additionally, the habit formation for older generations is lower compared to younger users which indicates an area for improvements to overcome this. Therefore, we should design an end-user feedback procedure to acquire the essential data to guarantee that no issues arise throughout the technology rollout.

4. Monitoring and Support
   24% of the survey responses indicated issues in the performance after rolling out the platform. As a result, the rollout framework should include appropriate monitoring and support procedures. The setup should ensure optimal HC performance and infrastructure setup is in place and agreed upon.

5. Rollout Framework Design

When compared to traditional application development, creating solutions with low-code/no-code platforms is presumed to be a speedy process. Agile approaches are ideal in this situation. Iterations can have a set of deliverables that are assessed rapidly after each iteration. Putting the agile framework in place necessitates taking into account large-scale organisations and teams all around the world. As a result, while adopting portfolio or program management, scaled agile frameworks can be considered.

Different Agile methods provide a pathway to scale Agile, especially for large projects. Below table lists the available methods and the criteria to assess the suitable framework [19].

Criteria	DAD	SAFe	LeSS 1	LeSS 2	Spotify	Nexus	RAGE
Team size	200 people or more. It also supports small and medium teams.	Large Enterprise includes more than 1 release trains (50 to 124 people in each release trains)	Up to 70 people or 10 SCRUM teams, 7 stakeholders in each team	Any large projects, More than thousand people on one product	Any large projects, Normally 250 to 300 people at Spotify (30 teams)	Three to nine SCRUM teams	No specific size but it support different size for enterprises
Training and certificate on	Workshops to explain the idea of DAD, Book of DAD is available	Training is needed and there should be certified, coaches	Seven companies in six countries are available for coaching	Seven companies in six countries are available for coaching	Lack of training	Scaled Professional SCRUM Training is needed	Training is conducted by webinar and presentation slides
Methods and practices adopted	Kanban Practices (mainly visualizing Work and limiting work in progress), SCRUM (almost all SCRUM practices), Agile Modeling which is the source for DAD's modeling and documentation practices, the Unified Process, XP, TDD and Agile Data.	SCRUM, Lean, Kanban, SCRUMban, DevOps and some practices of XP	SCRUM was fully adopted including additional practices for large projects	SCRUM was fully adopted including additional practices for large projects	Allow Kanban, SCRUM, DevOps and Lean Startup	SCRUM with additional practices in solving the dependency-related issues in multiple teams	Allow SCRUM, Kanban, Plan-Driven development and Hybrid approaches
Technical Practices required	High (Need to adopt many technical practices which require high technical skills)	Medium but should understand the use of portfolio management tools	Medium and low for SCRUM adopters	Medium and low for SCRUM adopters	Medium but teams should be able to communicate well	Medium and low for SCRUM adopters	Medium and low for SCRUM adopters
Organization Type	Multiple Organization and Enterprise practicality	Enterprises and portfolio level	Large Traditional organization	Enterprises	Enterprises specifically similar to Spotify	Portfolio level for medium project	Traditional and Agile Enterprises

Figure 19 Scaling Agile methods frameworks comparison [19]

We considered SAFe (Scaled Agile Framework) in this framework since it covers the large enterprises, supports commonly used methodologies and addresses the program and portfolio level. SAFe also has a considerable number of resources online that can be found under https://scaledagile.com.

The Citizen Developer is one of the low-code/no-code platform's major personas, and the following are essential recommendations for adopting a citizen development initiative that should be addressed in the framework [8]:

1. Automation to be considered on the individual and department level in a secure and scalable manner.
2. Small changes can lead to large-scale, impactful digital transformations.
3. Consider reusability, scalability, total cost of ownership, potential risks, and impact on an enterprise-level norms such as security.
4. Examine the ROI.
5. Consider coordinating, customizing, and monitoring the complete user experience across all channels at scale.
6. Ensure that citizen development capabilities play a more active part in the development process and understand how to identify the problems and solve them.
7. Establish a Citizen Development governance framework.

The rollout framework design will go through 6 stages that considers essential SAFe implementation:

1. Problem Definition
2. Technology Selection
3. Solution Setup
4. Develop Pilot Solution
5. Incremental Solution Rollout
6. Monitor and Support

In the rollout framework, we considered the strategic imperatives of digital transformation, which include digital resources, organisational structure, digital growth strategies, and metrics [17]. Digital resources include digital assets such as data storage and digital networking capabilities that bring together and match different users to meet their mutual requirements through digital methods. The organisational structure refers to a flexible structure that enables for the necessary organisational modifications to adapt to digital transformation. Digital growth strategies that take into account the platform's use and scalability. Finally, evaluate the metrics through KPIs that should be used to assess performance.

Below figure shows the steps taken in the process for the low-code/no-code platform rollout covering the six stages:

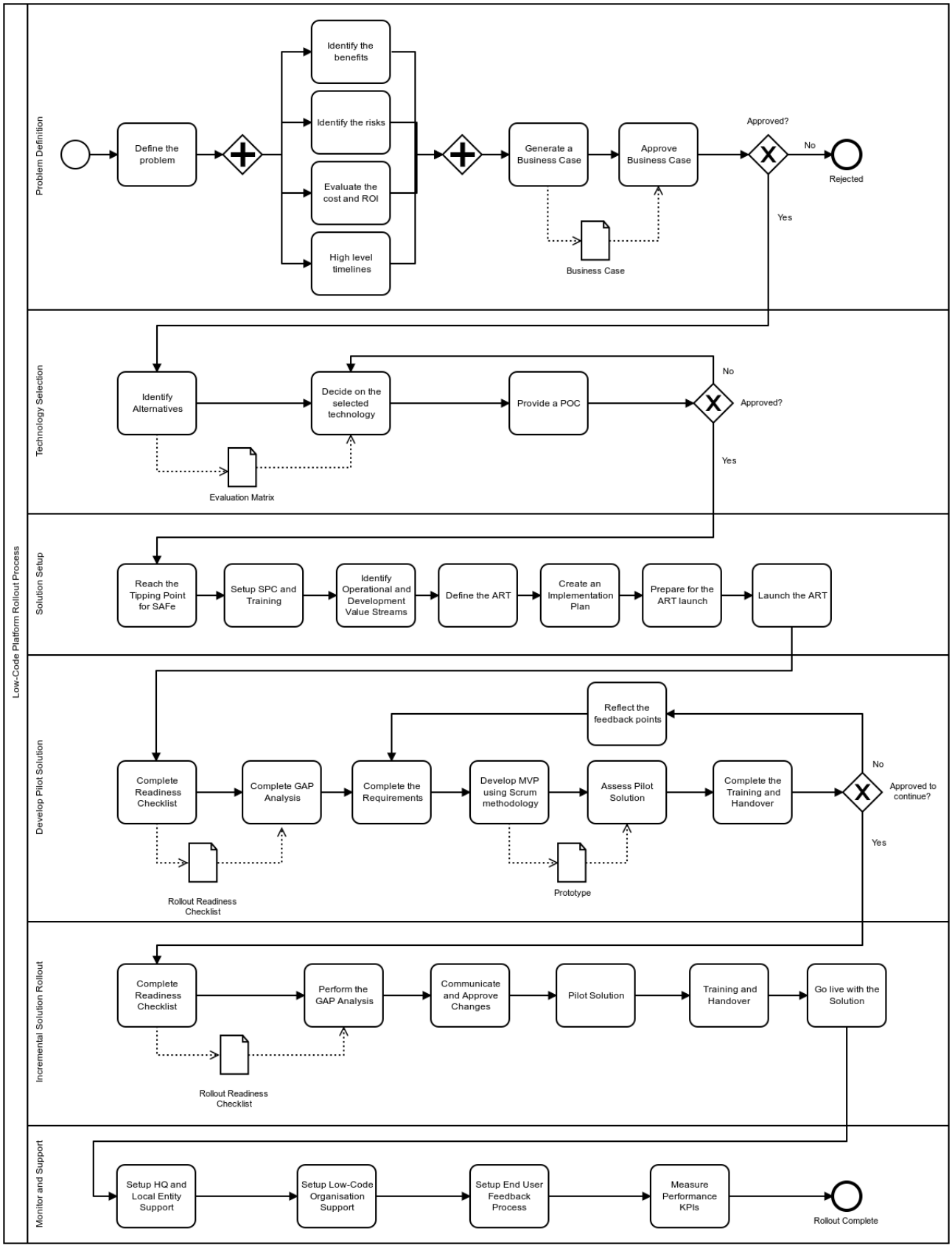


Figure 20 Low-code/no-code platform rollout process

The design of the rollout framework in details is as following:

5.1. Problem Definition

Defining the problem, as well as identifying features and benefits based on markets and user demands, is critical at this stage. All stakeholders must be included in the discussions, and the conversations and agreements must be in line with the organisation's strategy.

A Business Case must be created at this stage that details the benefits, risks, estimated costs, and return on investment, as well as the high-level timelines for implementing this platform. The details of the business case are based on the suggested alternatives that can be considered to cover the business need. Once the Business Case is approved, it is possible to proceed to the next stage.

5.2. Technology Selection

During this stage, several feasible alternatives are considered. The evaluation matrix should be used among the available options to decide on the best solution. Shortlisted solutions will need a Proof of Concept (POC) to be provided by the low-code/no-code platform organisation. The matrix was developed based on the interviews held with the organisations that use HC and others that did not proceed with selecting the solution.

Table 9 Low-Code/no-code technology selection aspects

No.	Aspect	Description
1	**User Interface**	• UI Capabilities and controls offered to end users • User experience for building forms • Easy onboarding process for builders • Styles provided • Responsive designs • Signature collections capability • Library of Workflow elements • The options for scripting low code • Multilingual
2	**Citizen Development**	• Maintainability of use cases for citizen developers • Training requirements • System reports issues and provides warnings in the architecture and design
3	**Cost**	• One-time, monthly, yearly • Operating Costs, Customizations and Training • Pricing Strategy used (Value-based pricing, Competitive pricing, Price skimming, Cost-plus pricing, Penetration pricing, Economy pricing, Dynamic pricing strategies or others) • Free trial availability
4	**Deployment**	• Mobile apps and supported Operating Systems • Deployment process and time to market • White label availability • Hosting On-premise or Cloud • Available solution environments: Development, Testing, or Production • Versioning possibility

5	**Integration**	• APIs • Robotic Process Automation (RPA) integrations • IoT Protocols • Barcode readers, NFC, Bluetooth devices and others • GPS APIs
6	**Data Management**	• Data Lake • AI and ML capabilities • BI tools • Data hosting and ownership • Encryption capabilities • Migration possibilities
7	**Performance and Monitoring**	• Monitoring and performance charts for developed forms • Logs and error reporting • Trends and patterns • KPIs setup • Availability and uptime of the low-code/no-code platform • Offline Mode with caching
8	**Reporting and Dashboards for End Users**	• Available dashboards • Report builders • Available formats for Export
9	**Security**	• User access management and permissions level • Secure authentication • Error handling • History logs • Backup • Archiving • GDPR compliance
10	**Scalability and Maintainability**	• Scale up resources • Load balance • Architecture sufficient for scaling including Global, regional and local entities management • Onboarding new and different types of users • Reusable Content such as forms or part of the forms • Impact of downtime • Data limitations • Maintenance, managing changes and versions • Software upgrades • Troubleshooting and Testability
11	**Organisation (Low-code/no-code Platform)**	• Expertise (Technology and Operations) • Use Cases and industry specifics • Location • Support and Customer Service • Project Management • Partnership with other organisation (isolated or engaged with other organisations) • Innovation (continuous upgrades, infrastructure, vision)

Each of the eleven aspects needs to have a weight that suits the organisation and a score assigned for the available solutions to decide on the shortlisted ones. The total weight for all aspects should be 100%, the Score is set as a value from zero to ten where ten indicates the highest score.

To calculate the value for each aspect, this formula is used: **(Score/10)*Weight**. Below figure illustrates a sample of the evaluation matrix and the results displayed in the web chart:

Aspect		Weight	Solution 1		Solution 2		Solution 3	
			Score	Value	Score	Value	Score	Value
1	User Interface	5%	10	5%	8	4%	7	4%
2	Citizen Development	25%	8	20%	8	20%	5	13%
3	Cost	15%	7	11%	9	14%	7	11%
4	Deployment	15%	7	11%	8	12%	6	9%
5	Integration	5%	10	5%	10	5%	8	4%
6	Data Management	5%	7	4%	7	4%	8	4%
7	Performance & Monitoring	5%	6	3%	8	4%	7	4%
8	Reporting & Dashboards	10%	9	9%	9	9%	8	8%
9	Security	5%	9	5%	10	5%	9	5%
10	Scalability & Maintainability	5%	10	5%	9	5%	10	5%
11	Organisation	5%	9	5%	7	4%	10	5%
	Total	**100%**		**81%**		**84%**		**65%**

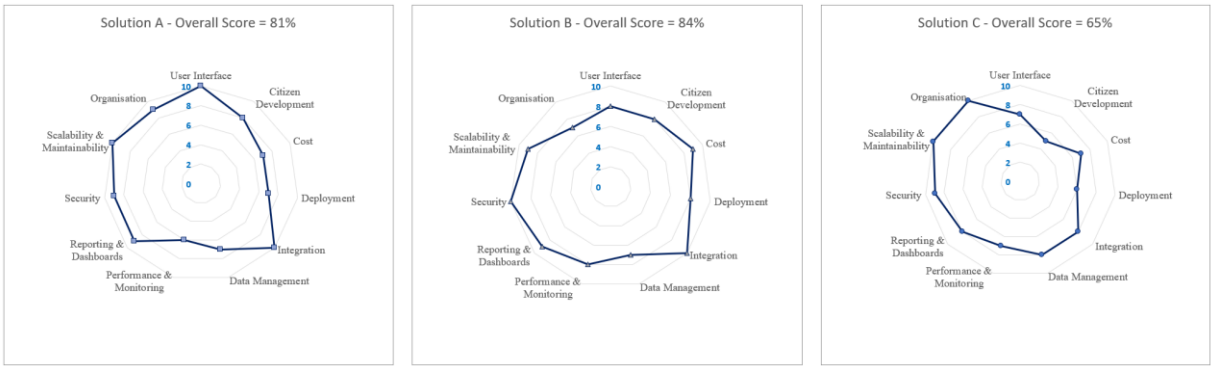


Figure 21 Technology selection evaluation matrix

5.3. Solution Setup

When compared to traditional application development, creating solutions with low-code/no-code platforms is a speedy process. Agile approaches are ideal in this situation. Iterations can have a set of deliverables that are assessed rapidly after each iteration. Putting the agile framework in place necessitates taking into account large-scale organisations and teams all around the world. As a result, while adopting portfolio or program management, scaled agile frameworks can be considered.

Different Agile methods provide a pathway to scale Agile, especially for large projects. Below table lists the available methods and the criteria to assess the suitable framework [19]. SAFe (Scaled Agile Framework) will be considered in this research since it covers the large enterprises, supports commonly used methodologies and addresses the program and portfolio level. SAFe also has a considerable amount of resources online that can be found under https://scaledagile.com.

SAFe Benefits and Challenges

According to a research of 52 unique organisations adopting SAFe, five from the scientific literature and 47 from the grey literature, the benefits of adopting SAFe are summarised in the table below [20]:

Table 10 Benefits of adopting SAFe [20]

Benefit	**Description**
Build in Quality	• Quality of product and releases • Defect Reduction • Continuous Improvement • Waste elimination by less duplicated work and reduced rework
Alignment	• Alignment between teams, tools, products and priorities • Collaboration between cross site and cross functional teams and diverse groups • Improved dependencies management • Vision is shared along with the goals and companywide strategy
Transparency	• Transparency in communication • Improved visibility
Organisational Benefits	• Productivity • Team Autonomy, more empowered teams • Improved employee engagement • Employee Satisfaction • Predictability in product delivery • Feedback
Business Benefits	• Cost controlled, reduced cost per epic and quality costs • Frequent and faster Deliveries • On time delivery • Responsiveness towards market and customer needs • Improved time to market • Increased Customer Satisfaction

The same research summarizes the challenges for adopting SAFe that need to be considered during the solution setup and implementation [20]:

Table 11 Challenges of adopting SAFe [20]

Challenge	**Description**
Organisational and cultural	• Change resistance • Difficult to implement agile mindset • Teams lacked autonomy • Plan driven or traditional culture
Roles	• Resistance for new roles • Staffing roles
Practices	• Defining values streams • First Program Increment Planning, chaotic, clash of time slots, lack of knowledge and technical dependencies topics. • Backlog Management and feature shaping • Test automation for legacy systems • Controversies with framework • Failure demand of Agile Release Train due to ineffective PSI, handling cross team dependencies across the ART's or rejection to take part into ART • Moving to SAFe feels like moving away from agile
Scaling and distribution	• Large and distributed settings including integrating non development units such as IT, HR and sales and marketing • GSD (Global Software Development) where critical gatherings were difficult due to distributed teams, deriving global priorities, different time zones and geographic distribution

The simplest way to implement SAFe is to use the framework's most basic configuration, Essential SAFe, which has the bare minimum elements required to succeed with SAFe.

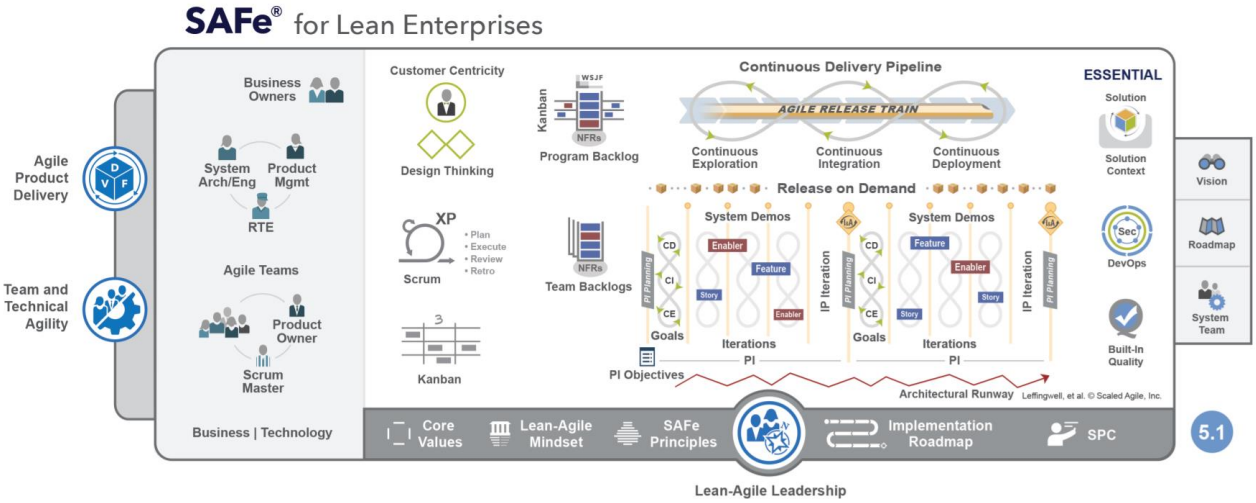


Figure 22 Essential SAFe configurations version 5.1 [21]

SAFe Implementation Roadmap

The solution setup includes the implementation roadmap that consists of the following steps:

1. Reaching the Tipping Point

The cause for the change to use SAFe, as well as the vision for it, should be established in this step. It is critical that the management is supporting the SAFe implementation, it is one of the main findings in the exploratory interviews that might cause a delay in the rollouts.

2. SPCs and Training

To begin adopting SAFe, certified SAFe Program Consultants (SPCs) are required. They are the change agents that act as consultants and put the structure in place. For low-code/no-code rollouts, one SPC would be sufficient for the implementation. The SPC can come from any department and have a variety of degrees of expertise, they can also be an outsourced consultant.

It is essential for the SPC to have the SAFe technical knowledge and experience to improve the existing processes. The SPC will assist the organisation through its business decision to go SAFe, establish an implementation strategy with organisational stakeholders, and then train executives, managers, and other leaders. The SPC should be part of the headquarter and covering the needs for implementing SAFe in the local entities.

In addition to the SPC, it is vital to train the resources to ensure that they have the necessary knowledge, skills, and tools to drive the change. All resources should be aware of the 10 SAFe Lean-Agile Principles in order to include them throughout the process. Executives, managers, and leaders should be given special consideration because they will be the ones to lead the organisation through the transformation process.

3. Identify the Operational Value Streams

In order to launch an ART (Agile Release Train), it is important to understand where the launch should take place. Financial reports could identify where the most value is and helps identifying

the operational value streams. This step should be kept simple and focused on the potential departments where low-code/no-code platform will be used. It includes identifying the key business processes, the departments supported, customers and targeted key business initiatives.

Figure 23 Sample for the Operational Stream Value

4. Identify the Development Value Streams

It is recommended that the initial decision be for a clear solution with a quick win to maximize the advantages and accelerate adoption. End-users who utilize the low-code/no-code platform's designed solution will become motivated users who will advocate the platform's use once they experience the benefits. This is one of the factors that will accelerate adoption. Once this is concluded the decision on where to launch the single ART can be made.

The Development Value Stream Canvas can assist in defining the value stream details of the selected first stream.

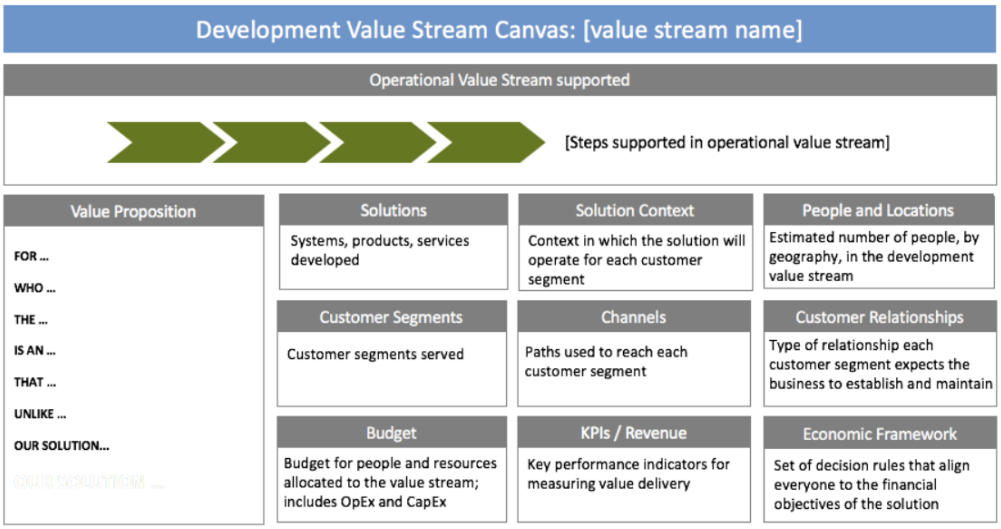


Figure 24 A Development Value Stream Canvas [21]

The Development Value Stream details the steps and people involved in developing the business solutions that operational value streams rely on. IT operations, legal, marketing, finance, support, compliance, security, and other departments are included. An example would be using the low-code/no-code platform to digitize the inspection checklist. One development stream is adequate when all steps in the operational value stream touch all solutions.

5. Define the ART

In this step, the ART is defined using the ART Canvas. It is recommended to pick one value stream with a quick win.

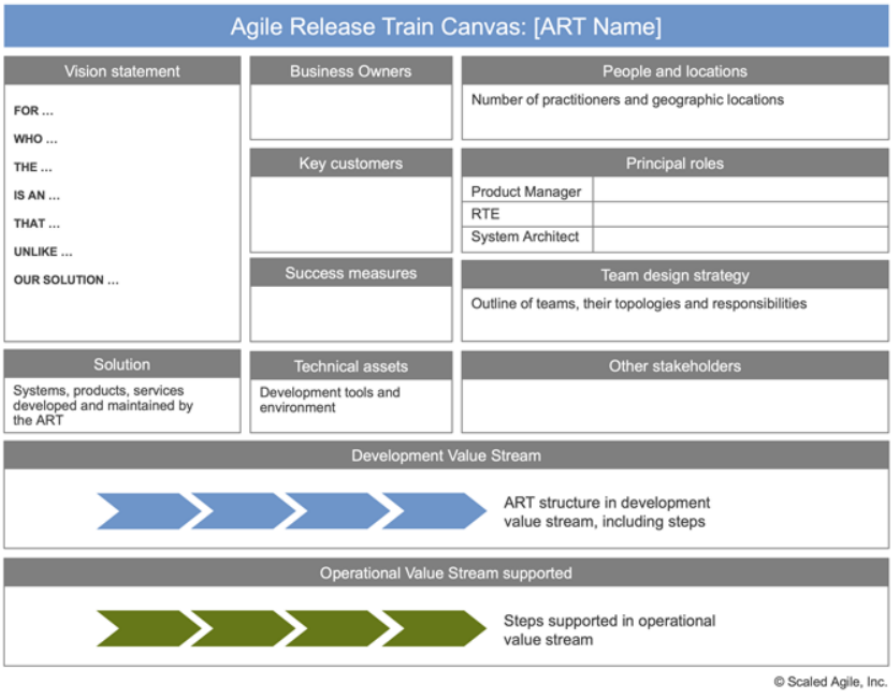


Figure 25 Agile Release Train Canvas [21]

The following roles need to be defined for the selected ART:

- RTE (Release Train Engineer): The RTE role is comparable to that of a Chief Scrum Master in terms of responsibilities. They follow up on the process that is set by the SPC. Since the resources for the low-code/no-code rollout are limited, one RTE is sufficient.
- Product Manager: A single product manager is sufficient to start the ART. The Product Manager covers the customer needs and develops the program vision.
- System Architect/Engineer: One Architect is required to define the technical and architectural vision. The Architect also defines the NFRs (Non-functional Requirements).
- Business Owner: Includes departments heads that are involved in the relevant ART from the headquarters in addition to the management in the local entities involved in this ART.

Each ART has multiple agile teams that follow the SCRUM framework. In the low-code/no-code implementation the teams can be based in the headquarter and the country or region where the digitized solution will be rolled out.

6. Create an Implementation Plan

Next step would be to create the Program Increment (PI) Roadmap. PIs are typically 8 to 12 weeks long. The most common pattern for a PI is four development Iterations followed by one Innovation and Planning (IP) iteration. At this step the following needs to be covered:

- Setup the PI Planning event
- Agree on the Capabilities, features and user stories setup using the relevant tools for managing projects and programs
- Identify the committed PI features and capabilities during the PI planning
- List the committed milestones and deliverables
- Set the forecasted milestones and deliverables

- Fill the forecasted PIs

Figure 26 Sample Implementation Plan

7. Prepare for the ART launch

The budget allocation and solution architecture should be discussed at this stage. Those factors were examined during the exploratory interviews and can have a detrimental impact if not adequately prepared for. The System Architect/Engineer should define the overall architecture for the system and identify the Non-functional Requirements (NFRs).

Product Management should modify the program backlog once the architecture is clear. The backlog is made up of features that demonstrate a company's desire to digitize a certain process using the low-code/no-code platform. Because the process may exist in several local entities, it must be prioritized and approved before it can be implemented following the Program Increment (PI) Roadmap. For each feature, personas, objectives, value, and scope must be determined.

The setup agreed using the relevant tool for the project and program management can utilize labels to indicate whether a feature is a global need or a requirement exclusive to a certain local entity. If relevant, similar labels can be used in the user stories. This will help in prioritizing the change's urgency and impact. Local entities can also go through the features and user stories and, if required, ask the Product Manager for changes. Depending on the necessity, the local entity or headquarters can handle the creation of forms using the low-code/no-code platform. If the form is utilized worldwide with a partial section for local entity purposes, it might also be created/modified by both.

The Agile teams should be established and the roles identified. The Agile team template can be filled to offer better visibility on the roles within the teams.

Table 12 Agile team template

Team Name	Role	Team Member	Location	Department
Team A	Scrum Master	First and Last Name	Country / City	
Team A	Product Owner			
Team A	Citizen Developer			
Team A	Solution Architect			

Team A	Tester			
Team A				

A decision should be made about who manages the low-code/no-code platform and who should be responsible, accountable, consulted, and informed about the changes. SAFe RACI template can be used for this purpose, however it might be complicated to start with it for the pilot solutions. The following is an example of the template that covers the ART and Agile teams roles but a simplified version should be developed:

Product Deliverables/ Activities	SAFe Level	Enterprise Executive/Business Owners/Customer	Product Manager	Release Train Engineer	Enterprise Architect	System and Solution Architect/Engineering	Product Owner	Scrum Master	Dev Team	DevOps & System Team	User Experience (UX)
Define Vision / Mission	Portfolio	AR	C	I	I	I	I	I	I	I	I
Define Strategic Themes	Portfolio	C	AR	I	C	I	I	I	I	I	I
Create product roadmap	Portfolio	C	AR	I	C	I	I	I	I	I	I
Create architectural strategy	Portfolio	I	I	I	AR	C	I	I	C	C	C
Create UX strategy and standards	Portfolio	I	C	I	C	C	I	I	C		AR
Fund ARTs	Portfolio	AR	R	R	C	I	I	I	I	I	I
Create Portfolio Business Epics	Portfolio		AR		C		C				
Create Portfolio Architectural Epics	Portfolio		I		AR	C	I		C	C	
Create Portfolio Operational Epics	Portfolio				R	C		I	C	AR	
Rank, maintain and communicate Portfolio Backlog	Portfolio		AR	I	C	I	I	I	I	I	I
Define Portfolio Metrics	Portfolio	I	AR	I	I	I	I	I	I	I	I
Create Business Features	Program		AR		C	R	R				C
Create Architectural Features	Program		I		C	AR	C		C		
Create Operational Features	Program				C	C		I	C	AR	
Rank, maintain and communicate Program Backlog	Program		R	AR	C	R	C	I	I	I	I
Define Non-Functional Requirements (NFRs)	Program		C	I	AR	R	C	I	C	C	I
Define Program PI Objectives	Program	I	I	AR	I	I	C	C	I	I	I
Participate in PI Planning	Program	R	R	AR	R	R	R	R	R	R	R
Communicate PI Plan	Program	I	I	AR	I	I	I	I	I	I	I
Solution architecture	Program			I	C	AR	C	I	C	C	I
UI/UX Design	Program		C	I	I	C	R	I	C	I	AR
Define Team PI Objectives (including stretch objectives)	Program			I		I	R	AR	R	C	C
Release schedule	Program	C	AR	C	C	I	I	I	I	C	I
Sprint Planning	Team			I		I	C	AR	R	R	R
Create stories	Team			I		C	AR	I	R	I	C
Refine story acceptance criteria	Team					I	C	C	AR	C	I
Rank, maintain and communicate Team Backlog	Team		I	C		C	AR	C	C	I	I
Track and manage the PI	Program		I	AR	I	I	I	I	I	I	I
Participate in Scrum of Scrums	Program		I	AR	I	I	C	R	C	C	C
Attend PI Demo	Program	R	R	AR	R	R	R	R	R	R	R
Define Program Metrics	Program			AR	C	C	C	C	I	I	I
Inspect & Adapt Workshop	Program		I	AR	C	C	R	R	R	R	R
Release on demand	Program	C	C	AR	C	I	C	C	I	C	I
End-to-end validation	Program					C	C	C	R	AR	C
Story development	Team						C	I	AR	C	C
Story validation	Team			I			C	C	AR	I	C
Execute IP Sprint	Team			I		I	C	AR	R	C	I
Sprint Retrospective	Team			I		I	C	AR	R	R	R
Maintain build, CI, tools, source repo, doc repo, other engineering assets	Program			I	C	C		C	C	AR	
Maintain deployment tools, scripts, environments	Program			I	C	C		C	C	AR	
Communities of Practice (Architecture Building Blocks) (Domain Architecture - Business, Integration, Data, Application, Security, and Infrastructure)	Program			I	R	AR	I	I	R	R	R

Figure 27 SAFe engineering roles and responsibilities [22]

The roles should cater for the relevant resources needed for the low-code/no-code platforms development and implementation. This includes:

- Admin: manages the low-code/no-code platform and could be present in the headquarters, local entities or regional.
- Builder: Consists of citizen developers or IT related resources that create the forms and manage the low-code/no-code platform development.
- Support: For post implementation and questions raised, those could be from the headquarters, local entities or from the low-code/no-code platform organisation.
- Business Analyst
- Data Analyst
- Technical Writers
- QA
- Information Architecture
- Security Specialist
- End-User Trainers

8. Launch the ART

This step starts with training the ART teams. Cost effective training needs to be considered due to the global teams involved. Once the training is performed, the PI Planning can start. The outcome of the PI planning covers the committed PI objectives and the program board with features delivery dates, milestones and including any dependencies.

Teams participating in the ART will utilize Scrum. Iteration Planning, Daily Standup, Iteration Review, Retrospective, and Backlog Refinement are among the team's events. It's critical to include end-users who will be using the produced solutions in backlog refinement and retrospective events.

A System Demo occurs once each iteration is completed. And at the end of the Program Increment, the Inspect & Adapt event takes place. The solution is demonstrated and evaluated by the ART. The Scrum of Scrum and PO Sync events can be merged into one event called ART Sync for low-code/no-code rollouts. The goal is to determine how effectively the ART is going in relation to the milestones and PI objectives.

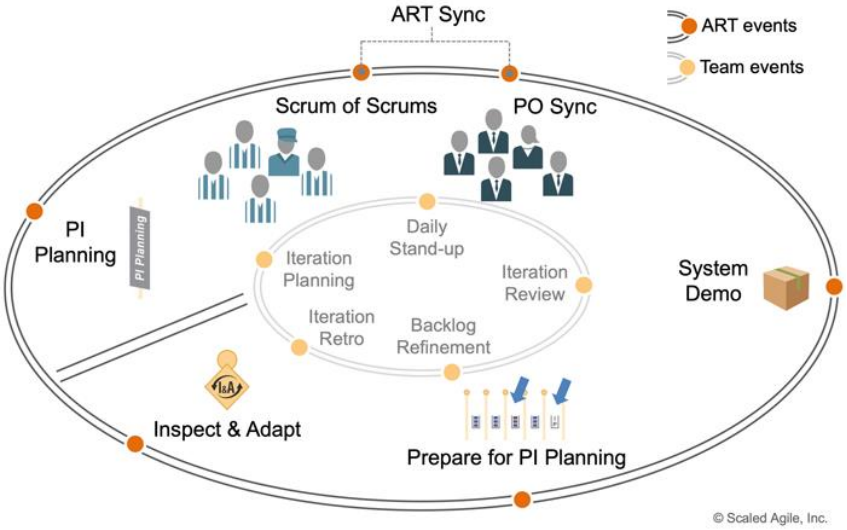


Figure 28 ART events [21]

9. Future Steps

Other steps can be taken after the successful PI deliverables that are optional and include:

- Launching more ARTs and Value Streams.
- Extending the solution on a portfolio level and accelerate.
- Introduce other SAFe related roles.
- Lean Budget Guardrails, which covers the policies and governance, can be considered. Participatory Budgeting is used to allocate funds to development streams in this approach.

Facilitating Factors

There are a number of factors to consider in addition to the SAFe implementation roadmap. The main facilitating factors include four levels: Human Resources, Tools, Workflow and Approvals, and Budgeting related items.

Human Resources

The following should be considered under Human Resources:

1. Resources and Skills: The people engaged should be able to develop the low-code/no-code platform and have the necessary skills. It should be determined at this point whether further resources are necessary for the SAFe and low-code/no-code platform setup.

2. Training: Any low-code/no-code platform training for admins, builders, and end users should be provided. This should take into account citizen developers as well as IT resources for complex solutions. If an organisational change is required as a result of the new process's digitalization, it should be considered prior to deployment.

3. Sharing Resources: Because low-code/no-code platforms aren't typically employed as core systems in organisations , project resources may be constrained. Sharing resources among agile teams is an interesting idea to examine. For example, all agile teams may share a Scrum Master under one ART. In this circumstance, part-time roles might also be explored.

4. Hub-and- Spoke Distribution Model: It is the ideal team distribution model for organisations with high local autonomy. The headquarters (the hub) develops core practices and distributes them to the local entities (spokes); on the other hand, spokes provide the hub with effective local practices to distribute to other spokes.

Tools

The existing tools to manage projects and programs should be examined. The new setup might require the purchase of new tools or additional plugins. It could also be possible that the existing tools can be utilized with some changes.

Workflow and Approvals

An approval mechanism should be developed for new requirements and changes to existing implementations, which can be initiated by local entities or the headquarters. To eliminate duplications and ensure that other entities benefit, any modifications should go via the ART. There should be a plan in place for notifying involved stakeholders about the new requirements and changes. It should also include information about the rollout plans.

Budgeting

The budget has an impact on the timeliness of the deployment and might be a show-stopper in some cases. As a result, it is critical to determine the expenditures associated with every program, as well as the costs allocated to each local entity. The budgeting should cover:

1. Any tools or plugins required
2. Hiring employees if needed
3. Certifications
4. Training
5. Customizations required by the low-code/no-code platform

6. The billing setup: the setup should be transparent and agreed upon with the low-code/no-code platform organisation. The setup should include the creation of the PO, reporting of billing efforts, invoices issued per project or per local organisation, and a single point of contact for follow-ups and invoice generation.

5.4. Develop the Pilot Solution

A checklist must be completed before starting any rollout in any entity. This will aid in identifying any more development that is necessary. It should also put a hold on the plans for deployment in that local entity until it is ready. The following items are on the checklist:

- Privacy and Security: Any blockers or prerequisites.
- Devices: The availability of the mobile devices, tablets, printers, laptops or other devices required for the implementation.
- Integrations: With other IT systems.
- Database: The structure, location, cleaning and migration.
- Resources: Availability for the implementation.
- Regulations: Any regulatory requirements, GDPR related items or blockers.

Table 13 Framework design - readiness checklist

No	Item	Ready?	Notes
1	Privacy and Security	Yes/No	
2	Devices	Yes/No	
3	Integrations	Yes/No	
4	Database	Yes/No	
5	Resources	Yes/No	
6	Regulations	Yes/No	

All local entities concerned should be consulted about the requirements. The development of the pilot solution should consider whether the process allows for localization or is a standard solution pushed by the headquarters. Using the sizing and PI roadmap is recommended to start with the simplest implementation. The Pilot can consist of one or more local entities depending on the maturity and feedback obtained and the ART decisions.

Before starting the process, it is advised that a prototype is created and shared with the stakeholders. The MVP approach should be used during the development. Following the completion of a sprint, retrospective meetings are required to review the changes made and provide the necessary feedback. The relevant training and handover procedure should be performed at the conclusion of the pilot.

5.5. Incremental Solution Rollout

After the pilot is successful and the feedback points are reflected, the incremental solution rollout begins. The next entities should go through the same process by filling the readiness checklist. A thorough gap analysis should be conducted, and the results should be communicated with the headquarters as well as the local entities concerned. Major modifications need the creation of a prototype that can be shared with the appropriate parties. Entities who have already implemented the solution should be included in the testing phase.

The required training and handover procedure should be completed at the conclusion of the rollout.

5.6. Monitor and Support

Monitor and support is the last stage of the rollout framework. In this stage, the support setup should be in place which covers the following:

1. HQ and Local Entity Support Setup:
   This setup should be established for the HQ support and to identify the contacts for the local entity support. It should also include the escalation process based on the severity of the issues reported.

2. Low-code/no-code Organisation Support Setup:
   An agreement should be in place on the method for the support provided by the low-code/no-code organisation. It should cover the working hours for the support team, the methods used and identify the main point of contacts.

3. End User Feedback Process:
   A proper setup should be in place for the feedback obtained from the end users after implementing a solution.

4. Performance KPIs:
   Based on the KPIs identified for the deliverables and ARTs, the performance KPIs should be calculated and reported. Performance KPIs should take into consideration the before and after implementation. This would enable the entities to realise the benefits of the digitization using the low-code/no-code platforms.

5.7. Discussion

The designed framework covers the principles that were obtained during the exploratory interviews. The framework provides clear steps within each stage that can be followed by the organisations. The management support and approval was taken into consideration in early stages of the rollout. We defined the required roles during the relevant stages, a clear process is in place for that.

On the other hand, the budget is assessed in the problem definition stage based on the suggested solutions. Training and handover was considered after completing a pilot or rollout. We also focused on having motivated users by implementing the low-code/no-code platform for processes that have a high benefit within the steps of identifying the operational and development value streams.

A checklist is evaluated for each rollout to guarantee that the appropriate entity is ready to start the low-code/no-code platform rollout. The checklist should solve the issues that occurred for some rollouts where devices or resources were unavailable.

During the monitoring and support stage, we took into account a variety of factors to ensure that a clear support structure is in place. It covers all bases and provides the necessary help for a smooth rollout. The support setup should address the concerns mentioned during the surveys

that are related to the performance of the low-code/no-code platform. In addition to the highlighted unclear support process in the exploratory interviews.

6. Evaluation of the rollout framework

6.1. Rollout Framework Design Evaluation Method

The suggested rollout framework was evaluated with the help of the COO and CEO of the low-code/no-code platform organisation HC. A PowerPoint presentation was used to deliver the framework to each individual separately, which can be found in Appendix D. The presentation began with a summary of the exploratory interviews and end-user survey results, as well as the hypothesis that influenced the development of the framework. The rest of the presentation focused on the stages of the framework that have received feedback.

The feedback was gathered using the criteria outlined by Riemenschneider et al. in a technology acceptance article [23]. The article examined five existing models of individual acceptance of information technology tools in the domain of individual acceptance of methodologies. The models covered are: Technology Acceptance Model (TAM), TAM2, Perceived Characteristics of Innovating (PCI), Theory of Planned Behavior (TPB) and Model of Personal Computer Utilization (MPCU).

The feedback was measured using the following constructs: usefulness, ease of use, compatibility, and behavioural intention. Each construct was assessed using a Likert Scale ranging from 1 to 5, with 1 indicating strong disagreement, 3 indicating neutrality, and 5 indicating strong agreement. Additional comments were obtained in addition to the constructs.

The details of the constructs include:

- Usefulness: Measures the usefulness of having the steps or checklist. It refers whether having that stage adds value and improves the process of rolling out the low-code/no-code platform.
- Ease of use: Measures if the steps are clear, understandable to follow and do not require a lot of efforts.
- Compatibility: Measures whether the stages fit well with the way the organisations that use the low-code/no-code platform work.
- Behavioural Intention: Measures the intention to use of the low-code/no-code platform by the organisation that attempts to use their platform.

		1	2	3	4	5
Construct	**Question**	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
Usefulness	Having this stage/step is useful					
Ease of use	The steps are clear and understandable					
Compatibility	Using these steps fits well with the way the organisations work					
Behavioural Intention	Do you expect organisations that are attempting to use low-code to want to use it?					
Comments:						

Figure 29 Feedback measurement constructs

Other constructs were left out for a variety of reasons. Because the feedback did not include organisations using HC, several constructs could not be evaluated. Other factors include the absence of use of the framework, which means that some constructs cannot be measured.

6.2. Rollout Framework Evaluation Feedback

For each stage of the rollout framework, feedback was gathered. Because the solution setup stage is large, numerous steps were measured separately in this stage.

Stage 1: Problem Definition

In this stage, the problem definition flow was measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	4	4	3	4
Person 2	5	5	5	5

- Usefulness: The two participants agreed that this stage is beneficial to the rollout process.
- Ease of Use: According to the two participants, the steps in this stage are simple to use and follow.
- Compatibility: One of the participants rated the compatibility as neutral. Stating that it is suitable for large organisations, but might not be for smaller organisations.
- Behavioural Intention: The two participants agreed that organisations that are attempting to use low-code/no-code will use it.

Additional Comments received from the participants:

- Cost evaluation may be difficult at this point since the alternatives have not yet been determined.
- SaaS delivery should be rapid and agile; this is a vital step to have, but some may overlook it.
- At this stage, the vision should evaluate if the solution may benefit the organisation in other demands or challenges.

Stage 2: Technology Selection

In this stage, the technology selection steps and selection matrix were measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	4	4	4	4
Person 2	5	5	5	5

- Usefulness: The two participants agreed that this stage is beneficial to the rollout process.
- Ease of Use: The two participants agreed that the steps in addition to the technology selection matrix are simple to use and follow.
- Compatibility: The two participants agreed that this stage is suitable for organisations that will use the low-code/no-code platforms.
- Behavioural Intention: The two participants agreed that organisations that are attempting to use low-code/no-code will follow the steps within this stage.

Additional Comments received from the participants:

- The matrix should consider the scope or focus of the business need. It should not be measured solely on the basis of its technological capabilities.

Stage 3: Solution Setup (Step 1 – Step 4)

In this stage, the first 4 steps for the solution setup using SAFe are measured. The steps include: Reaching the Tipping Point, SPCs and Training , Identify the Operational Value Streams and Identify the Development Value Streams. The 4 steps were measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	2	2	3	2
Person 2	5	3	3	3

- Usefulness: The usefulness construct received several types of feedback. While one person firmly believed that having and following it is beneficial, the other participant disagreed. One participant's disagreement stems from the fact that purchasing a low-code/no-code platform needs a simple and rapid solution.
- Ease of Use: This construct was rated as neutral and disagreement. Low-code/no-code platforms are marketed as a solution that is bought and setup without the overhead and extra requirements. Therefore having those steps in place are not easy to follow.
- Compatibility: Compatibility received a neutral rating from the two participants. It is appropriate for large organisations that have implemented SCRUM and are searching for a scaled agile framework, but it is not appropriate for other organisations .
- Behavioural Intention: Ratings received are neutral and disagreement. Extra requirements are not required at the start of the low-code/no-code platform implementation. Organisations that do not intend to adopt a scaled agile framework, the likelihood of using it is minimal.

Additional Comments received from the participants:

- Easy integrations are usually expected from low-code/no-code platforms.
- Creating unnecessary demands for the process is not preferred.

Stage 3: Solution Setup (Step 5)

In this stage, step 5 for defining the ART is measured. This step covers using the ART canvas based on the selected value stream in addition to defining it. It was measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	4	4	4	4
Person 2	5	5	5	5

- Usefulness: This step was rated as useful for organisations that will invest in SAFe.
- Ease of Use: Defining the ART including the roles was rated as easy to use and follow.
- Compatibility: This step fits large organisations that have SCRUM in place or are willing to implement SCRUM and SAFe.

- Behavioural Intention: The intention to follow this step is high for organisations that will use SAFe.

Additional Comments received from the participants:

- Some roles such as the architecture could be involved at a later stage after the pilot is completed.

Stage 3: Solution Setup (Step 6 – Step 7)

In this stage, step 6 covers the creation of an Implementation Plan while step 7 covers the preparation for the ART launch. The steps were measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	4	3	4	3
Person 2	4	4	3	3

- Usefulness: Having the implementation plan and the preparation for launching the ART was rated as useful and beneficial.
- Ease of Use: For organisations that don't need to follow SCRUM or SAFe, this stage will not be easy to use.
- Compatibility: This step fits large organisations that have SCRUM in place or are willing to implement SAFe. Some organisations might not require the Scrum master or other roles when the low-code/no-code platform offers ready use cases that can be implemented immediately.
- Behavioural Intention: The intention to follow this step is neutral depending on the use cases and the methodology that the organisations want to invest in.

Additional Comments received from the participants:

- Some agile teams roles might not be required such as end user trainers since the platform is self-explanatory. In other cases it could be needed.
- Technical writers might not be required since low-code/no-code platforms should have a simplified UI where what you see is what you get.
- RACI requires a lot of work; the less the efforts, the better. It could be useful after the pilot, when details are more evident.
- Big companies might take 70% of these steps.

Stage 3: Solution Setup (Step 8)

In this stage, step 8 covers the launching of the ART. This step was measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	4	4	4	4
Person 2	5	5	4	4

- Usefulness: the steps to launch the ART are beneficial for organisations that decided to follow SAFe.
- Ease of Use: The steps are easy to follow on the ART level and team level.

- Compatibility: This step fits large organisations that have SCRUM in place or are willing to implement SAFe.
- Behavioural Intention: The intention to follow this step is high depending on the methodology that the organisations want to invest in.

Additional Comments received from the participants:
- ART sync is necessary to guarantee that milestones are on track and objectives are met.

Stage 4: Develop the Pilot Solution

This stage was measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	**4**	**4**	**4**	**4**
Person 2	**5**	**5**	**5**	**5**

- Usefulness: The participants agreed that the steps to fill the readiness checklist, going through the ART events, producing the prototype, training and handover are beneficial for the rollout process.
- Ease of Use: The participants agreed that the steps are easy to follow and use.
- Compatibility: Having the steps and readiness checklist is compatible with the way the organisations work.
- Behavioural Intention: The intention to follow this step is high.

Additional Comments received from the participants:
- The checklist requires a new option “Not applicable”.
- Confirm the business case again at this stage as it might have changed.

Stage 5: Incremental Solution Rollout

This stage was measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	**4**	**4**	**4**	**4**
Person 2	**5**	**5**	**5**	**5**

- Usefulness: The participants agreed that the steps are beneficial for the rollout process.
- Ease of Use: The participants agreed that the steps are clear and easy to follow.
- Compatibility: The participants agreed that the steps are compatible with the way the organisations work.
- Behavioural Intention: The intention to follow the steps in this stage is high.

Additional Comments received from the participants:
- At this stage, the architecture can be involved to cover the scalability aspect if they were not considered at the beginning.
- Different pilots may take different approaches; at this point, the choice on how to proceed can be made.

Stage 6: Monitor and Support

This stage was measured as following:

	Usefulness	Ease of Use	Compatibility	Behavioural Intention
Person 1	**4**	**4**	**4**	**4**
Person 2	**5**	**5**	**5**	**5**

- Usefulness: The participants agreed that the monitoring and support in different levels are beneficial for the rollout process.
- Ease of Use: The participants agreed that the setups for the monitoring and support are clear and easy to follow.
- Compatibility: The participants agreed that all levels included in this stage are compatible with the organisations that will be using the low-code/no-code platforms.
- Behavioural Intention: The intention to follow the steps in this stage is high.

No additional comments were received from the participants regarding this stage.

6.3. Evaluation Feedback Summary

Based on the participants feedback, the proposed framework is beneficial for large organisations that are willing to implement SCRUM and essential SAFe. Otherwise the framework is heavy and adds another layer that is not helpful for low-code/no-code platforms.

The overall feedback received shows issues in stage3 where essential SAFe implementation is required. This is mainly for small organisations and those who do not wish to invest in SAFe. Other stages show strong agreements and positive feedbacks. Below figures show the average feedback received for the stages for each measurement construct.

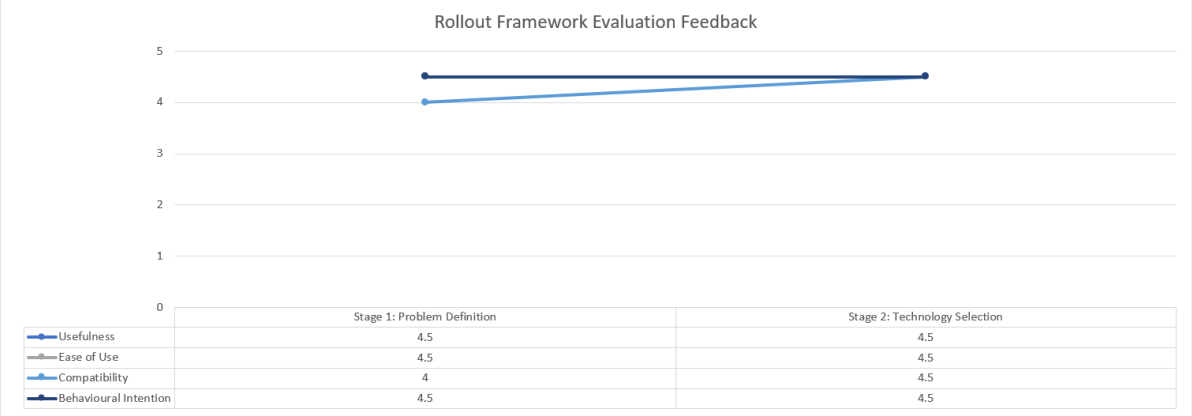


Figure 30 Rollout framework evaluation feedback

To conclude, low-code/no-code platforms will necessitate two frameworks that may be utilized dependent on the demands of the organisation. The first framework is a simple framework that takes minimal effort and resources. The second framework is based on essential SAFe framework for organisations that want to implement a broader scaled agile framework. The next sections will go through the two frameworks in more details.

7. Refinement of the Rollout Framework design

Based on the feedback evaluation of the rollout framework, we concluded that the low-code/no-code platforms can be rolled out using one of the two refined frameworks. The decision is based on different elements that are summarized in below table:

Table 14 Elements for choosing the right framework

Factor	Simplified Framework	Framework using Essential SAFe
Organisation Size	Small, medium or large.	Large
Requirements	Implement ready-made use cases or creating forms for limited processes.	Creating multiple forms for different processes with cross-functional teams.
Efforts	Minimal	Setup requires efforts until it is mature.
Cost	No extra costs.	Might include extra costs for the consultancy or tools.

According to Mergel et al's research [18], both frameworks should address organisational culture, as well as personnel capabilities and mindset, in order for the digital transformation to be sustainable.

7.1. Simplified Framework

The simplified framework is used as a light weight framework. It goes well with organisation that don't want to invest in a process or resources to rollout the low-code/no-code platforms.

The framework goes through the following 7 stages illustrated in below figure:

1. Problem Definition
2. Technology Selection
3. MVP Definition
4. Roles Definition
5. Develop the Pilot Solution
6. Incremental Solution Rollout
7. Monitor and Support

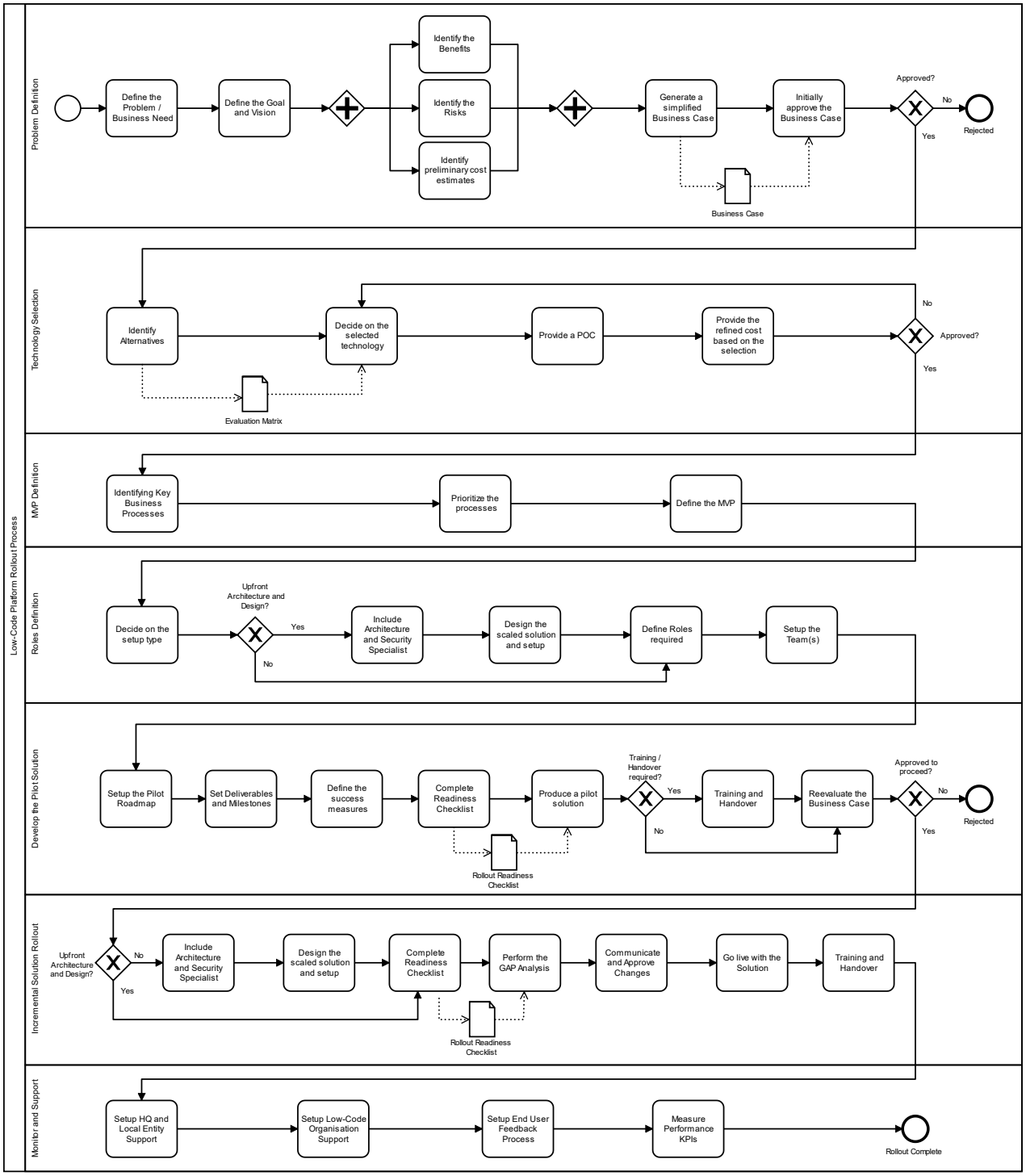


Figure 31 Simplified rollout framework

The design of the rollout framework in details is as following:

1. Problem Definition

The problem and business need are identified at this stage. Other benefits that can be obtained in addition to covering this business need are discussed. This information should be used to outline the project's goal and vision. At this point, it is important to realise the short-term and long-term benefits that need to be realised.

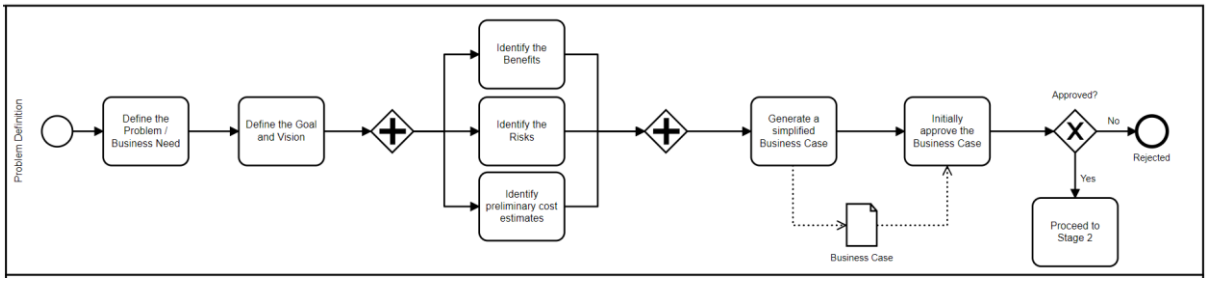


Figure 32 Simplified framework - Problem definition stage

This stage features a simplified business case that just requires the most basic information in order to proceed forward with the approval. The benefits, risks, and preliminary cost estimates based on the offered solutions and technologies make up the business case. Because high-level estimations are only required at this stage, the cost can be set as a range.

This stage is critical for gaining management support for the proposed solutions. If not secured, it is one of the most essential issues that will affect the rollout in the future.

2. Technology Selection

The Technology Evaluation Matrix is used to identify and evaluate the potential solutions that might be considered to meet the business need (Appendix E). The weights should be filled according to their value to the company. At this point, the evaluation is focused on the emphasis area and should not be limited to software elements.

A POC can be requested from the high-performing low-code/no-code organisation once the overall score for the solutions has been determined. If the organisation will be employing the ready-made use cases, the POC can be bypassed.

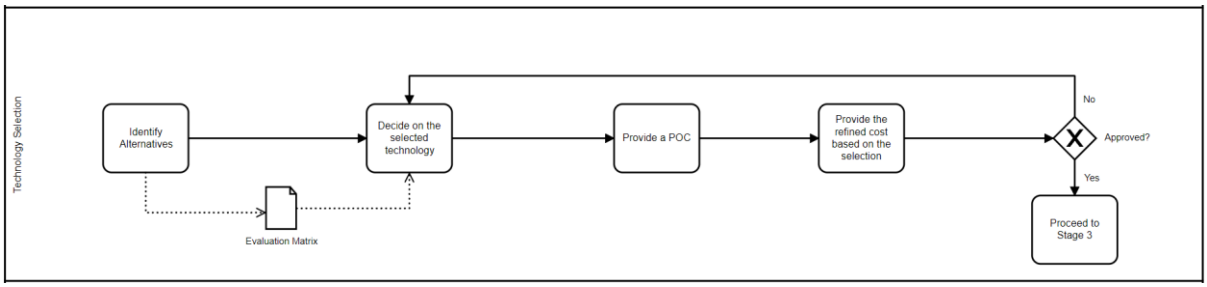


Figure 33 Simplified framework - Technology selection stage

The cost should be adjusted depending on the chosen solution at the conclusion of this stage. The following stage can begin once the cost has been authorized by management.

3. MVP Definition

This stage begins with identifying the key business processes where low-code/no-code platforms can be implemented. It is important to understand where the launch should take place. Financial reports could identify where the most value is. This step should be kept simple and focused on the potential departments where low-code/no-code platform can be used.

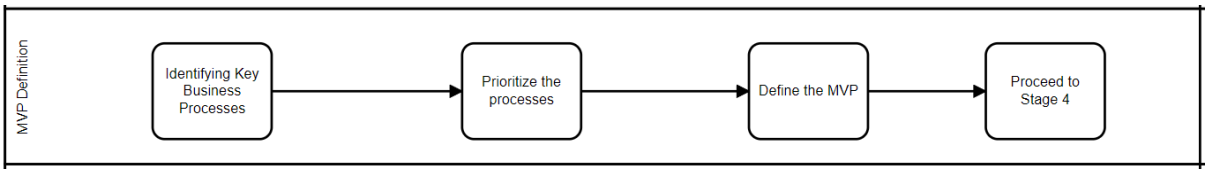


Figure 34 Simplified framework - MVP definition stage

It is recommended that the initial decision be for a clear solution with a quick win to maximize the advantages and accelerate adoption. End-users who utilize the low-code/no-code platform's designed solution will become motivated users who will advocate the platform's use once they experience the benefits. This is one of the factors that will accelerate adoption.

4. Roles Definition

Before defining the roles involved, it must be decided whether the solution's architecture and design will be considered at this point. If the design and architecture are critical, the appropriate security experts and architects should be consulted. They will be in charge of designing a scalable and secure solution. At this point, there is a need to discuss how the requirements are captured and classified as regional, local, or headquarter-related requirements.

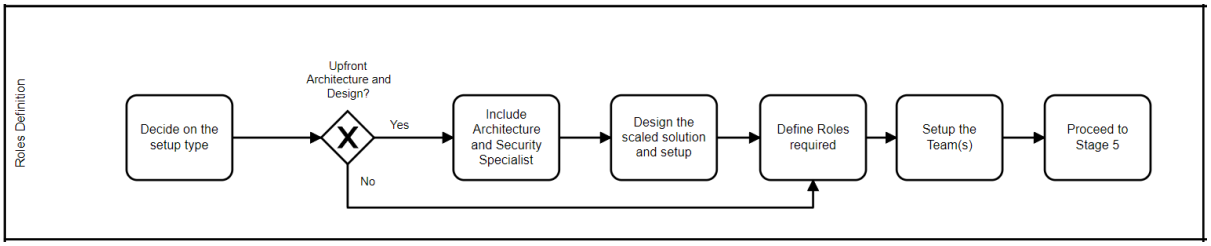


Figure 35 Simplified framework - Roles definition stage

A design is not necessary in all circumstances. It's possible that this is because the low-code/no-code platform already includes such capabilities. It might also be related to implementing the low-code/no-code platform's ready-made use cases.

Other roles are defined according to the needs of the organisation. The roles that might be involved in the process are listed below:

- Project Manager: To manage the project implementation.
- Business Owners: Involved from different departments and countries based on the need.
- Business Analyst: For complicated requirements where the process is unclear.
- Admin/Super User: To manage the administration of the low-code/no-code platform.
- Builder: The user from the IT or the citizen developer who creates the forms and workflows.
- Support: Users who provide the needed support during the rollout of the low-code/no-code platform.

- Data Analyst: This user might be required to provide the necessary input regarding the data management aspect.
- Technical Writers: The low-code/no-code platforms should be self-explanatory, however, some organisations might require to document the technical aspects of the projects.
- QA: Involved when processes developed are complex and require sufficient testing.
- End-User Trainers: For new processes that don't exist before or require some additional training.

To ensure a successful rollout, the organisation should assess whether certain skills or changes in employee mindset are necessary. The team(s) should be set up and ready to go on to the next stage after the roles are specified using the table below and the skills are covered.

Table 15 Team roles template

Team Name	Role	Team Member	Location	Department
Team A	Scrum Master	First and Last Name	Country / City	
Team A	Product Owner			
Team A	Citizen Developer			
Team A	Solution Architect			
Team A	Tester			
Team A				

5. Develop the Pilot Solution

The pilot roadmap is established at the start of this stage. The necessary entities and teams will be involved based on the MVP's priorities. For the initial MVP, clear deliverables and milestones should be established..

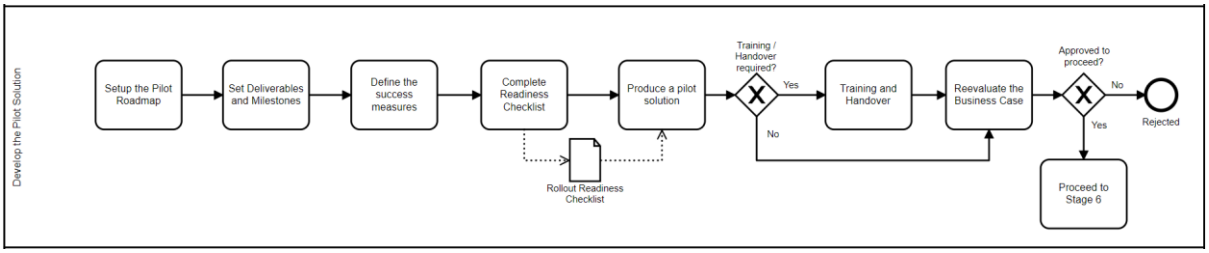


Figure 36 Simplified framework - Develop the pilot solution stage

At this point, the success criteria should be established. It will aid in determining if the pilot was successful. The readiness checklist should be completed before starting the pilot.

Table 16 Develop the pilot solution stage - Readiness checklist

No	Item	Ready?	Notes
1	Privacy and Security	Yes/No/ Not Applicable	
2	Devices	Yes/No/ Not Applicable	
3	Integrations	Yes/No/ Not Applicable	
4	Database	Yes/No/ Not Applicable	
5	Resources	Yes/No/ Not Applicable	
6	Regulations	Yes/No/ Not Applicable	

A training and handover may be necessary when the pilot solution is completed. It is not always required since the solution is self-explanatory and does not need additional training.

The appropriate MVPs will be evaluated at the conclusion of this stage, and the business case will be updated with the new information. To advance to the next stage, management approval is necessary. The solution will be rejected otherwise.

6. Incremental Solution Rollout

The incremental solution rollout should begin when the pilot solution has been approved and all feedback points have been considered. Security experts and architects should be consulted based on the preceding determination if architecture and design are not necessary upfront. They will be responsible for creating a scalable and secure solution.

Since it was not covered before, it is now necessary to examine how requirements are collected and categorised as regional, local, or headquarter-related requirements.

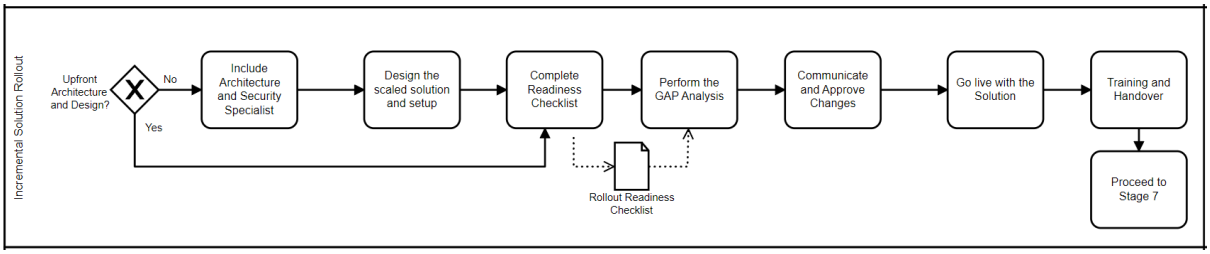


Figure 37 Simplified framework - Incremental solution rollout

The same readiness checklist used in the pilot should be completed before moving forward with the rollout. Any additional needs or changes will be considered when a gap analysis is completed. Depending on the adjustments that need to be made, relevant teams will be involved.

After completing the rollout and going live, a training and handover might be needed.

7. Monitor and Support

Monitor and support is the last stage of the rollout framework. In this stage, the support setup should be in place which covers the following:

1. HQ and Local Entity Support Setup:
   This setup should be established for the HQ support and to identify the contacts for the local entity support. It should also include the escalation process based on the severity of the issues reported.

2. Low-code/no-code Organisation Support Setup:
   An agreement should be in place on the method for the support provided by the low-code/no-code organisation. It should cover the working hours for the support team, the methods used and identify the main point of contacts based on the severity.

3. End User Feedback Process:
   A proper setup should be in place for the feedback obtained from the end users after implementing a solution.

4. Performance KPIs:
   Based on the KPIs identified for the deliverables and ARTs, the performance KPIs should be calculated and reported. Performance KPIs should take into consideration the before and after implementation. This would enable the entities to realise the benefits of the digitization using the low-code/no-code platforms.

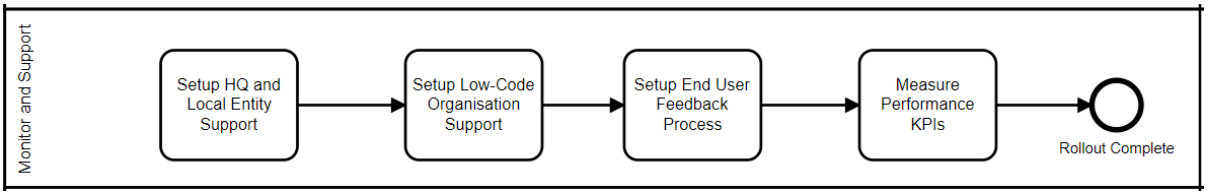


Figure 38 Simplified framework - Monitor and support stage

7.2. Framework using Essential SAFe

The framework designed using essential SAFe is used for organisations that want to invest in Scrum and scaled agile to rollout the low-code/no-code platform. Based on the feedback provided, the framework designed under Chapter 6 should consider the following changes in each stage:

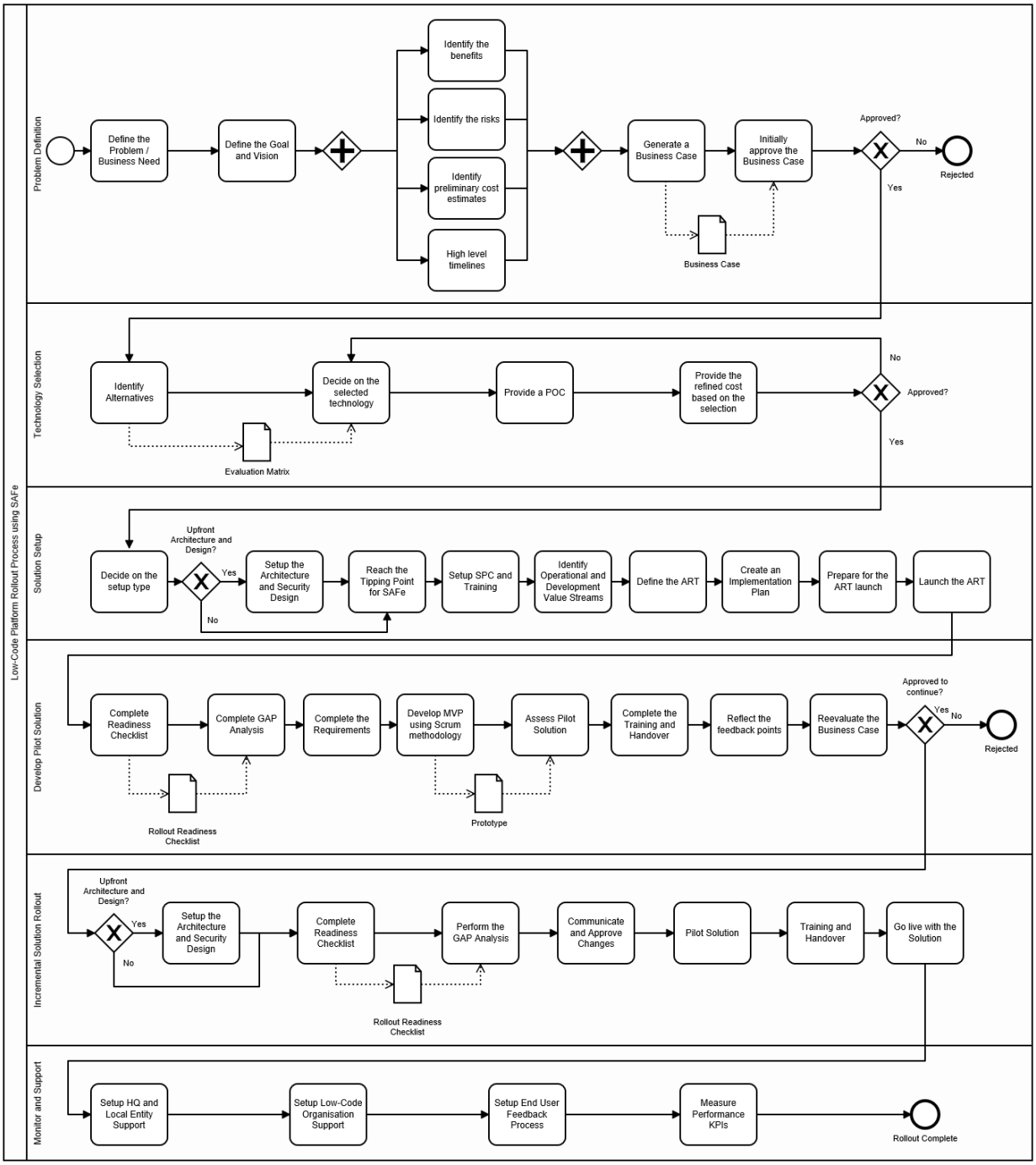


Figure 39 SAFe based rollout framework

1. Problem Definition

When solving a problem or covering a business need, the organisation should think about how the solution may benefit other areas. Because the cost is difficult to quantify at this point, the framework should factor in the preliminary cost.

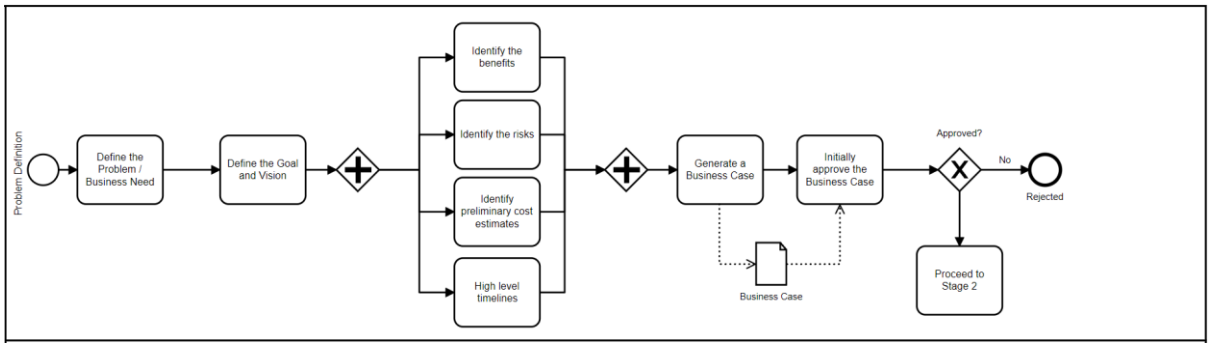


Figure 40 SAFe based framework - Problem definition stage

2. Technology Selection

The evaluation selection matrix should be based on the scope of work to be covered. It should not simply look at the solution from a technological standpoint only. For example, if the goal of the low-code/no-code platform selection is to automate inspections, the available use cases and expertise for the low-code/no-code platform organisation weigh in on the low-code/no-code platform's technology aspects.

The chosen solution should allow the organisation to come up with a refined cost that should be considered for approval at the conclusion of this stage.

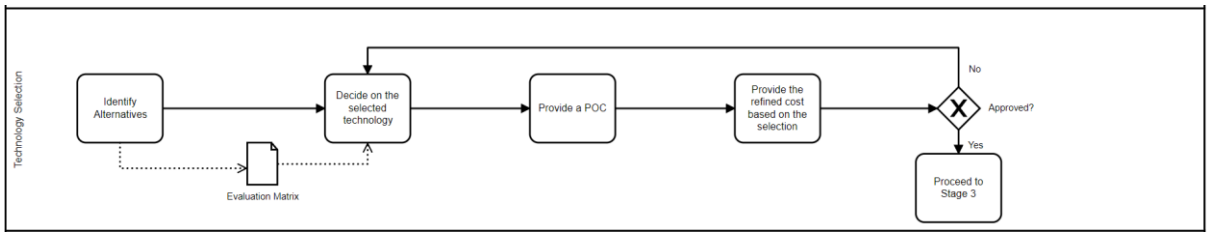


Figure 41 SAFe based framework - Technology selection stage

3. Solution Setup

This stage should determine whether the setup should consider the security and architecture design upfront. The organisation should decide whether they want to have the design upfront and plan the ARTs accordingly, or whether they want to analyse and examine the security and architecture design once the first pilot is completed.

If the setup will employ ready-made use cases that do not demand additional work, the ART's setup and launch should take into account a lightweight implementation.

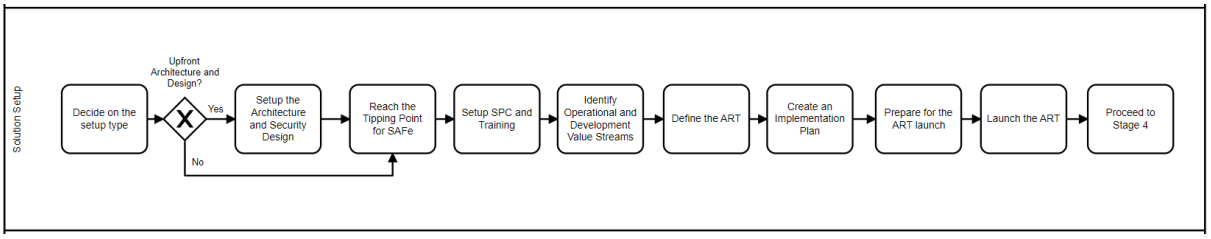


Figure 42 SAFe based framework - Solution setup stage

4. Develop the Pilot Solution

Filling out the readiness checklist is required at this stage in order to continue with the pilot. If any component of the checklist is not relevant to the pilot, the option "Not Applicable" should be included.

Table 17 Develop the pilot solution stage - Readiness checklist

No	Item	Ready?	Notes
1	Privacy and Security	Yes/No/ Not Applicable	
2	Devices	Yes/No/ Not Applicable	
3	Integrations	Yes/No/ Not Applicable	
4	Database	Yes/No/ Not Applicable	
5	Resources	Yes/No/ Not Applicable	
6	Regulations	Yes/No/ Not Applicable	

Depending on how the low-code/no-code platform is implemented, training and handover may be unnecessary. If a basic procedure is automated that does not necessitate any modifications to the existing process, training is not required and may be covered in a single session or by steps that end users can follow.

The business case should be re-evaluated by the organisation at the end of this stage before moving on to the next stage.

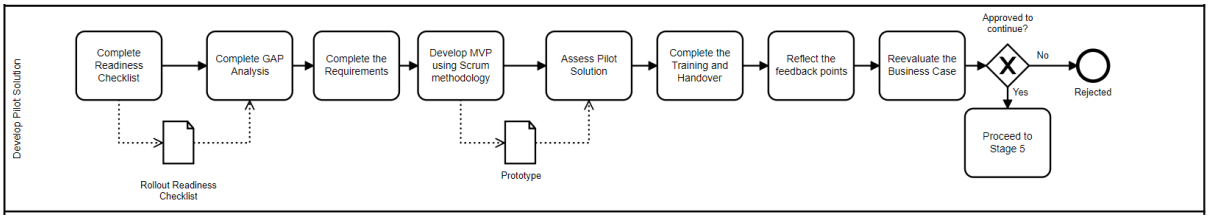


Figure 43 SAFe based framework - Develop the pilot solution stage

5. Incremental Solution Rollout

The decision taken for the upfront setup of security and architecture needs to be considered. If the design did not take into considerations the security and architecture, the organisation should consider them at this stage.

Similar to the pilot stage, training and handover could be optional depending on the needs.

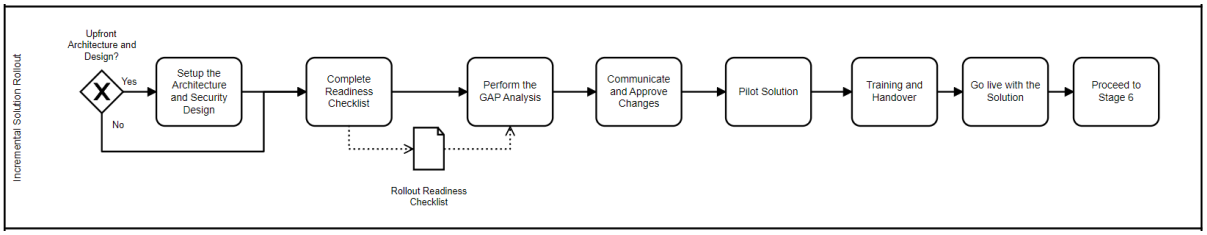


Figure 44 SAFe based framework - Incremental solution rollout stage

6. Monitor and Support

No change is required for this stage.

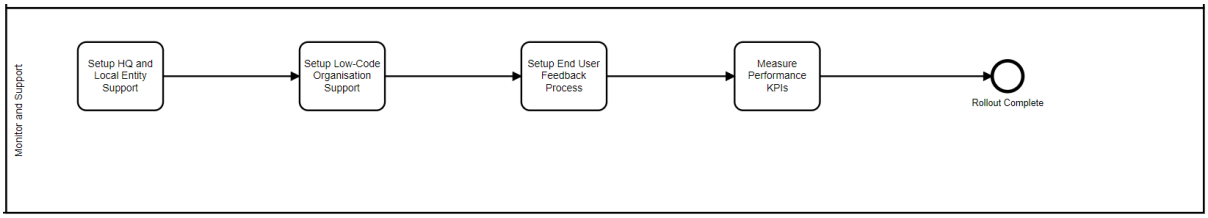


Figure 45 SAFe based framework - Monitor and support stage

7.3. Evaluation of the refined Framework

The two frameworks encompassed the principles identified during the exploratory interviews, as well as the end user survey findings and the evaluation of the proposed framework. Both frameworks provide organisations with clear steps within each stage that can be followed easily.

At several phases of the frameworks, management support was taken into account. This ensures that management is aware of the low-code/no-code platform's benefits and is on board with the rollout. It also covers the prioritization for the key areas that the business can make use of the low-code/no-code platforms. During the relevant stages, where a clear procedure is in place, we also specified the essential roles and responsibilities.

Low-code/no-code platforms are required to capture data and guarantee that error rates are kept low. The architecture design and security in the frameworks take into account the data management structure as well as any required interconnections with other systems or dashboards. As part of the requirements and gap analysis, this is also included.

Scalability is another important aspect of the frameworks that may be explored before or after the pilot is done, depending on how the architecture is set up. As a result of the discussions held throughout those stages, reworks would be minimized. If necessary, customizations will be noted.

In the frameworks, the cost is evaluated in several stages depending on the POC and pilot solution. The budget allocation approval should take into account the entities involved and examine the entire cost of ownership (TCO).

For each rollout, a checklist is evaluated to ensure that the right entity is ready to begin the low-code/no-code platform rollout. This includes any regulatory constraints, resource allocation, privacy and security concerns, required devices, and system integrations.

When necessary, training and handover were considered. Because the automated process is so simple, certain rollouts may not require end-user training. Others, on the other hand, should be considered to ensure that they have the necessary conditions to perform. By adopting the low-code/no-code platform for processes with a high benefit, we also concentrated on having motivated users.

The monitoring and support stage entails establishing a clear support system that covers all bases and provides the necessary help for a successful launch. This includes the availability of support as well as the performance of the low-code/no-code platform. A feedback process is also required to collect input from users of various ages and expertise levels. This can help to discover any areas of improvements to take the appropriate corrective action.

8. Discussion

This chapter covers the key interpretations for the research along with the limitations.

8.1. Interpretation of the results

We began the research with the literature review in order to have a better grasp of the digital transformation and the barriers or impediments to implementing new technologies. While some of the findings from the exploratory interviews are relevant to those publications, we discovered additional factors that are unique to HC rollouts. Although certain factors are specific to HC rollouts, the majority of them are probably applicable to other low-code/no-code platforms.

For existing rollouts, we used the UTAUT2 model to assess HC adoption and acceptability. The results demonstrated a good level of acceptance, with some findings that might help improve the rollout process in the future. The UTAUT2 model was straightforward to use and covered the needs. Is it possible, though, that if more users answered the survey, different findings would emerge? We find it unlikely to produce significant findings since our research covered most aspects. Is it possible that certain data indicate better technology adoption depending on the location of the rollout? This could be a factor that can be further analysed in future work.

We designed the framework considering essential SAFe, however we discovered afterwards that it is too difficult and heavy to deploy in some organisations. The two refined frameworks are intended to solve this issue. Do you think the simplified framework will be used more often? Or will organisations be willing to invest in SAFe in order to scale solutions across departments or regions? We find it likely that organisations who already invest in Scrum or SAFe, will adopt the SAFe based framework. However, organisations who aren't going to invest in SAFe won't think about it for low-code/no-code rollouts and may stick with the simplified framework.

8.2. Limitations

This section highlights the limitations of this research, including:

- **Exploratory interviews number**
  We conducted the exploratory interviews with 14 interviewees. Even though we have limited number of respondents, we saw saturation at the tenth interview, therefore we have some reason to believe we could be complete.

- **Exploratory interviews diversity**
  The emphasis of the exploratory interviews was on a single organisation that had existing rollouts. Other organisations participated in the interviews, and their input was valuable in the development of the framework. However, their focus was on the problem definition stage and technology selection since they did not proceed with HC selection. If other organisations that utilize HC were included, additional challenges and success factor could have been identified that were not addressed in the study.

- **Survey responses**
  We sent the survey to 530 users, but only 10% of them responded anonymously. Further discussions were necessary in certain circumstances to have a deeper grasp of the issues. While the number of respondents would have been too few if our objective had been to do statistical tests to reject or accept hypotheses, but in fact we were just looking of interesting insights to help us build the rollout framework, so the 53 respondents was quite enough.

- **UTAUT model validation**
  We did not test the UTAUT model statistically as it was not one of the objectives. Furthermore, because the data was limited, the UTAUT hypothesis was not tested.

- **Framework validation**
  With the help of the HC management team, we validated the framework. The validation process did not include any clients. The use of the rollout framework in actual HC implementations may have shown other viewpoints that aided in the frameworks' design. This was not validated because of time restrictions.

9. Conclusion

This chapter discusses the main findings, the research's contributions, the answers to the research questions posed in the introduction chapter, and potential future work.

9.1. Summary of Findings

We concluded that several factors have a positive or negative impact on the low-code/no-code platform rollout speed. We identified these factors through the exploratory interviews which fall under 3 contexts: technology, organisation and external environment contexts. The main positive factors include the management support from the organisation that wants to rollout the platform in addition to the data management, and transparency and cost savings.

On the other hand, unclear roles and responsibilities, as well as budget allocation, the unclear process, and the cost are all factors that have a negative impact on the rollout speed. Another factor that might hinder the implementation is the lack of adequate support from the low-code/no-code organisation.

We also measured the satisfaction of the existing rollouts using a survey. The end-user survey measured a variety of constructs, including performance satisfaction, effort required to use HC, social influence, and facilitating conditions, as well as motivation, habit formation, and intention to use HC. As a result, the framework included several considerations: user training, conveying the vision of rolling out the low-code/no-code platform to teams, developing an end-user feedback mechanism, and assuring effective monitoring and support setup.

A single framework is insufficient for all organizations to rollout the low-code/no-code platform. The appropriate framework to use is determined by the demands of the company, as well as the structure and size of the company. We developed a simple framework for the organisations that are looking for a framework that can be implemented rapidly with limited resources. We also developed another framework for organisations who are willing to invest in a program-level SAFe implementation.

As a result of our research, we were able to make the following contributions:

- We identified the organisation, technology and external environment context factors that influence the pace with which low-code/no-code platforms are deployed.
- We developed a survey that may be used to assess end-user satisfaction and low-code/no-code platform adoption using the UTAUT2 model.
- We designed an evaluation matrix that may be used to aid in the evaluation and selection of low-code/no-code platforms.
- We identified the four levels of the facilitating factors for the rollouts that cover: Human Resources, Tools, Workflow and Approvals, and Budgeting related items.
- We designed two frameworks to rollout the low-code/no-code platform. A simplified, light-weight framework, as well as another SAFe based framework.
- We created the readiness checklist that can be utilized before starting any low-code/no-code platform rollout.
- We identified the setups required for a successful monitoring and support stage that cover the HQ and local entity, low-code/no-code organisation support, end user feedback process, and performance KPIs.

9.2. Answers to the research questions

The following research questions were answered within this research as following:

Question 1: What can improve the speed of adoption for low-code/no-code applications?

We identified the factors that affect the low-code/no-code platform rollout speed using exploratory interviews. The findings were classified into three contexts: organisation, technology and external environment contexts. Details are available under "Exploratory Interviews Findings."

The main negative factors are:

- The organisation context: factors include the budget allocation, unclear or absent roles and responsibilities, unclear process and requirements, knowledge transfer, and human resources unavailability.
- The technology context: the high cost of advanced development using the platform, as well as the expenses of customisations and total cost of ownership. In addition to the security policies to follow, the devices' connection, and availability, and the complex solutions that may require a complicated architecture and design.
- The external environment context: the factor related to the lack of adequate support from the low-code/no-code organisation or the platform limitations.

On the other hand, the organisation can focus on the main positive factors that can improve the rollout:

- The organisation context: the main factors include the management support, motivated users and existing knowledge of how to utilize applications by resources.
- The technology context: the factors include the data management and system integrations, as well as cost savings in terms of recruiting and safety.
- The external environment context: In particular circumstances, the regulatory requirements might push for a faster rollout.

Question 2: Would offering a rollout method and template help?

Based on the feedback received during the evaluation of the rollout framework that we designed, having the framework in place should assist organisations in efficiently adopting and rolling out low-code/no-code platforms. We developed two refined frameworks to address all of the issues raised in the exploratory interviews and end-user survey.

The frameworks also feature a technology assessment matrix for identifying the organization's needs and comparing them to the available solutions from various vendors in order to select the optimal solution.

Question 2: Does the low-code/no-code platform technology affect the speed of adoption?

We covered this question in both the exploratory interviews and the findings of the end user surveys. In the event of performance concerns, limited support, or platform limitations, the low-code/no-code platform's technology has an impact on the rollout speed. As a result, we recommend filling out the Technology Evaluation Matrix to guarantee that the needs are identified early on in the implementation.

9.3. Future Work

This section contains recommendations for future work that may be done to improve this study:

- **UTAUT test**

The UTAUT hypothesis was not tested due to time constraints. Having a larger sample size for end user responses and subsequent rollouts gives a statistically testable hypothesis.

- **Test the research hypothesis**

The following hypothesis can be tested after implementing one of the rollout frameworks:

  - H1: Management support for low-code/no-code platforms increases the likelihood of a faster rollout.
  - H2: Organisations that report defined roles and responsibilities but do not have a process in place have a delayed rollout.
  - H3: The high expenses of training and customisation are projected to slow the adoption of low-code/no-code platforms.
  - H4: Immediate implementation of pandemic-related health rules has an impact on the rollout.

- **SAFe setup**

Additional details can be added under the rollout framework that uses essential SAFe.

- **Evaluation of the refined rollout frameworks**

Further work can be done to evaluate the two rollout frameworks using organisations that want to adopt the low-code/no-code platforms. The organisations can choose the appropriate framework that suits their need.

10. Bibliography

[1] Richardson, C., & Rymer, J. R. (2016). Vendor landscape: The fractured, fertile terrain of low-code application platforms. FORRESTER, Janeiro.

[2] Barton, N., & Sahoo, S. (2021, October 29). Impact of Low Code on Tech Consulting and Outsourcing Services. Gartner. https://www.gartner.com/en/documents/4007733

[3] Sanchis, R., García-Perales, S., Fraile, F., & Poler, R. (2019). Low-Code as Enabler of Digital Transformation in Manufacturing Industry. Applied Sciences, 10(1). https://doi.org/10.3390/app10010012

[4] Definition of Citizen Developer - Gartner Information Technology Glossary. (n.d.). Gartner. https://www.gartner.com/en/information-technology/glossary/citizen-developer#:%7E:text=A%20citizen%20developer%20is%20an,or%20function%20other%20than%20IT.

[5] Hevener A. (2007), A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems, 19(2):87-92

[6] Collatto, D. C., Dresch, A., Lacerda, D. P., & Bentz, I. G. (2017). Is Action Design Research Indeed Necessary? Analysis and Synergies Between Action Research and Design Science Research. Systemic Practice and Action Research, 31(3), 239–267. https://doi.org/10.1007/s11213-017-9424-9

[7] Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/mis0742-1222240302

[8] Carroll, Noel & Móráin, Liam & Garrett, Dave & Jamnadass, Arjun. (2021). The Importance of Citizen Development for Digital Transformation.

[9] Tijan, E., Jović, M., Aksentijević, S., & Pucihar, A. (2021). Digital transformation in the maritime transport sector. Technological Forecasting and Social Change, 170, 120879.

[10] Yeh, H. T. (1995). The challenge of change—Opportunities and obstacles in deploying information technology. Journal of Systems Integration, 5(1), 5-21.

[11] Attaran, M. (2004). Exploring the relationship between information technology and business process reengineering. Information & management, 41(5), 585-596.

[12] Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, *27*(3), 425–478.

[13] Samaradiwakara, G D Manoja & Chandra, Gunawardene. (2014). Comparison of existing technology acceptance theories and models to suggest a well improved theory/model. International Technical Sciences Journal. 1. 21-36.

[14] Momani, A. M. (2020). The Unified Theory of Acceptance and Use of Technology. International Journal of Sociotechnology and Knowledge Development, 12(3), 79–98. https://doi.org/10.4018/ijskd.2020070105

[15] Wieringa, R. J. (2014). Design Science Methodology for Information Systems and Software Engineering (2014th ed.). Springer.

[16] Venkatesh, V., Thong, J. Y., & Xu, X. (2012). Consumer acceptance and use of information technology: extending the unified theory of acceptance and use of technology. MIS quarterly, 36(1), 157-178.

[17] Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Qi Dong, J., Fabian, N., & Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. Journal of Business Research, 122, 889–901. https://doi.org/10.1016/j.jbusres.2019.09.022

[18] Mergel, I., Edelmann, N., & Haug, N. (2019). Defining digital transformation: Results from expert interviews. Government Information Quarterly, 36(4), 101385. https://doi.org/10.1016/j.giq.2019.06.002

[19] Alqudah, M., & Razali, R. (2016). A review of scaling agile methods in large software development. International Journal on Advanced Science, Engineering and Information Technology, 6(6), 828-837.

[20] Putta, A., Paasivaara, M., & Lassenius, C. (2018, November). Benefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review. In International Conference on Product-Focused Software Process Improvement (pp. 334-351). Springer, Cham.

[21] SCALED AGILE. (2021, February). SAFe configurations. Scaled Agile Framework. https://www.scaledagileframework.com/

[22] Chebbi, D. (2018, March 7). SAFe Engineering Roles and Responsibilities (Developed using Scaled Agile Framework & SFIA). LinkedIn. https://www.linkedin.com/pulse/safe-engineering-roles-responsibilities-developed-using-deepak-chebbi/

[23] Riemenschneider, C., Hardgrave, B., & Davis, F. (2002). Explaining software developer acceptance of methodologies: a comparison of five theoretical models. IEEE Transactions on Software Engineering, 28(12), 1135–1145. https://doi.org/10.1109/tse.2002.1158287

11. Appendix

Appendix A: Exploratory Interview Form

The exploratory interview form had different categories that help in classifying the topics and allows further points to be raised. Filling the form depended on the points raised by the interviewee.

Background Information	• Age • Gender • Education Level and Background • Experience in implementing similar platforms • Total years of experience
HC Implementation	• Their Role in implementing HC • The idea behind implementing HC • Challenges during the implementation • The experience after implementing the software
Vision and Strategy	• The vision of the implementation and the supported strategy
Social Influence	• The management thinks that you should use HC • Other units are using HC • Competitors have similar tools
Finance	• Cost reduction after implementing HC • Benefits of using HC are higher than the cost • The price affects the intention to use HC
Resources	• Layoffs and job security • Flexible reallocating resources, job designs and organisational structure • Encourage developing new skills for resources • Any resistance to change from the employees • Diversity in resources affected the implementation • Any investments in employees for the implementation
Stakeholder Management	• Engagement and needs vs expectations • Cross functional communications
Technology	• Encouraging the use of technology • Other technologies used along with HC
Data Management	• Data Transparency • Data usages
Culture	• Tolerate failures • Encourage risk taking
Regulatory	• Regulations to use similar platforms • Government financial support for automations
Global vs Local Issues	• Any issues they faced during the rollout
Other points raised	

Appendix B: End user survey

Use of HC Software

A survey to better understand the adoption of low-code/no-code platforms. The responses will be utilized for a Leiden University research project.

All of the information that you provide will be treated as confidential and used solely for research purposes.

Your age

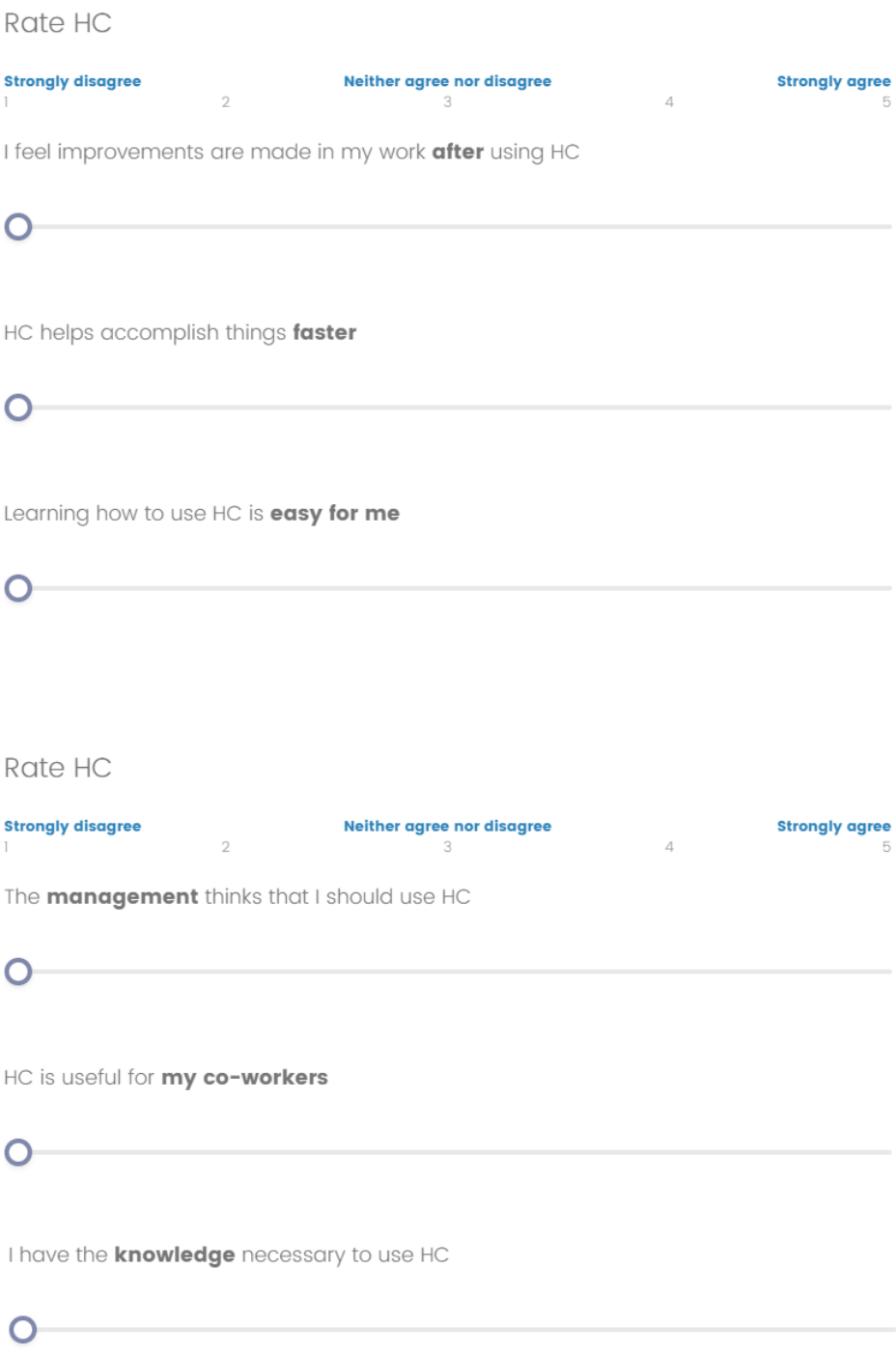
Your gender

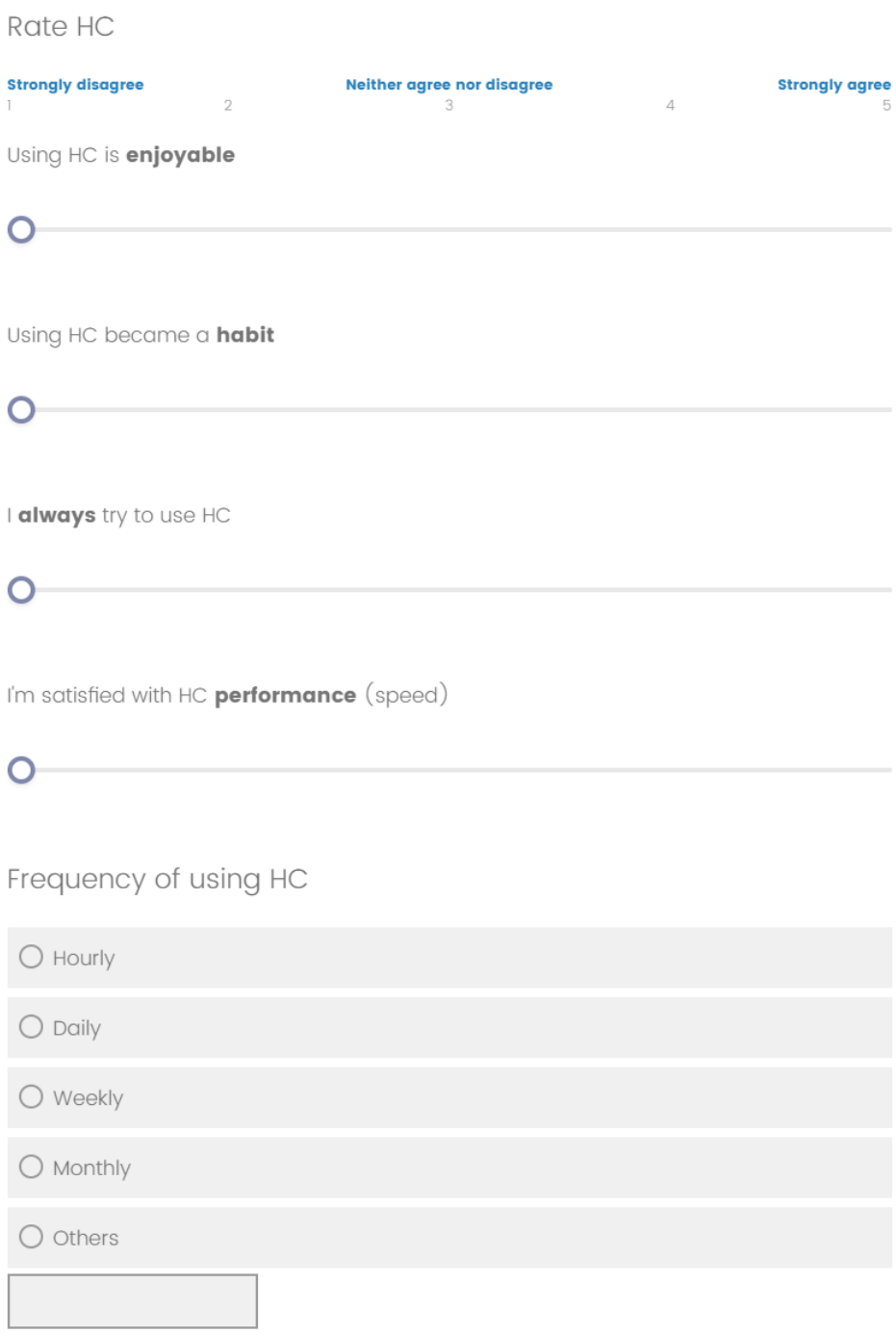
- ○ Female
- ○ Male
- ○ Other

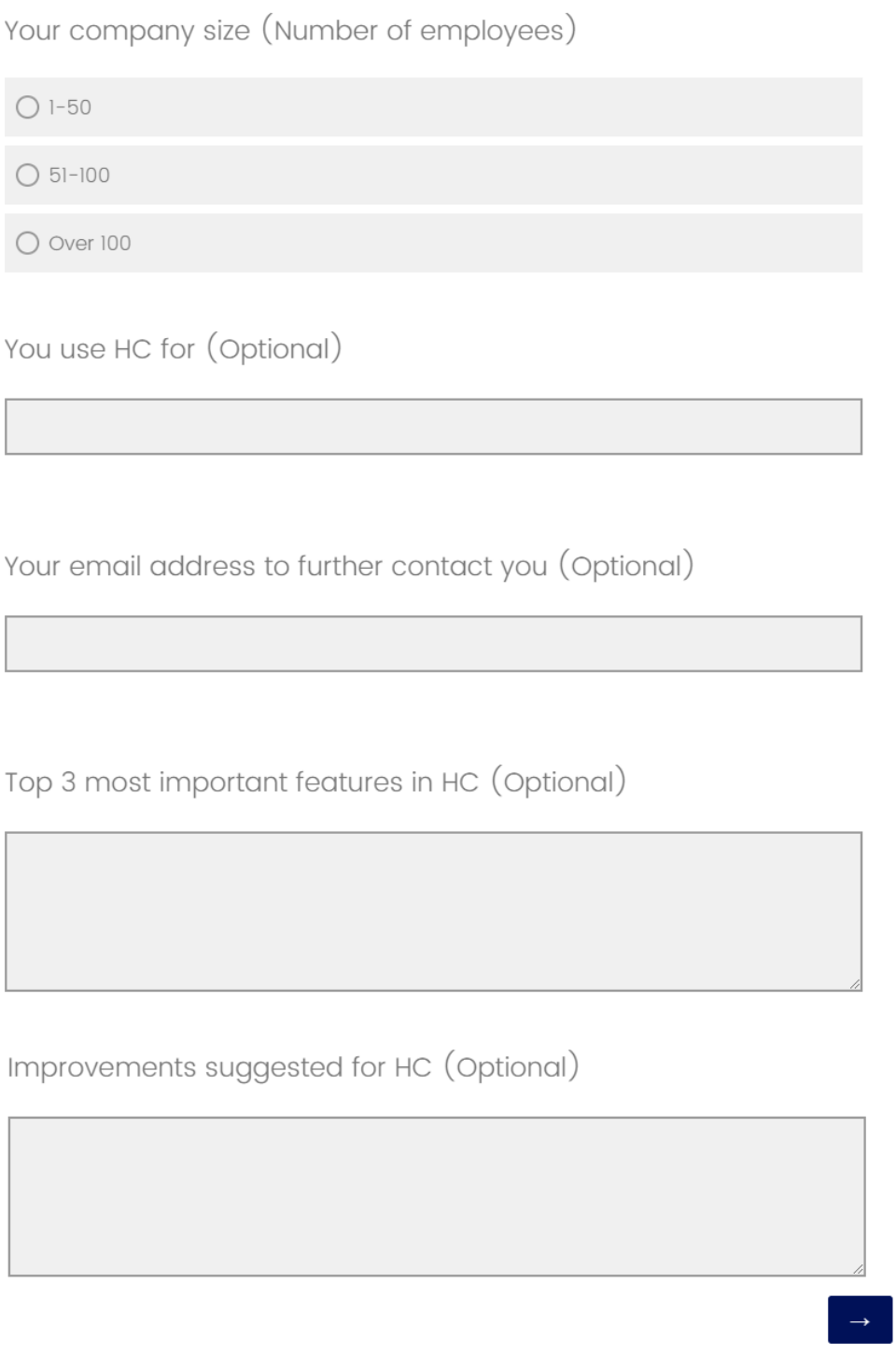
Your highest education degree

- ○ High School
- ○ University or College Degree
- ○ Others

Your total work experience *(in years)*

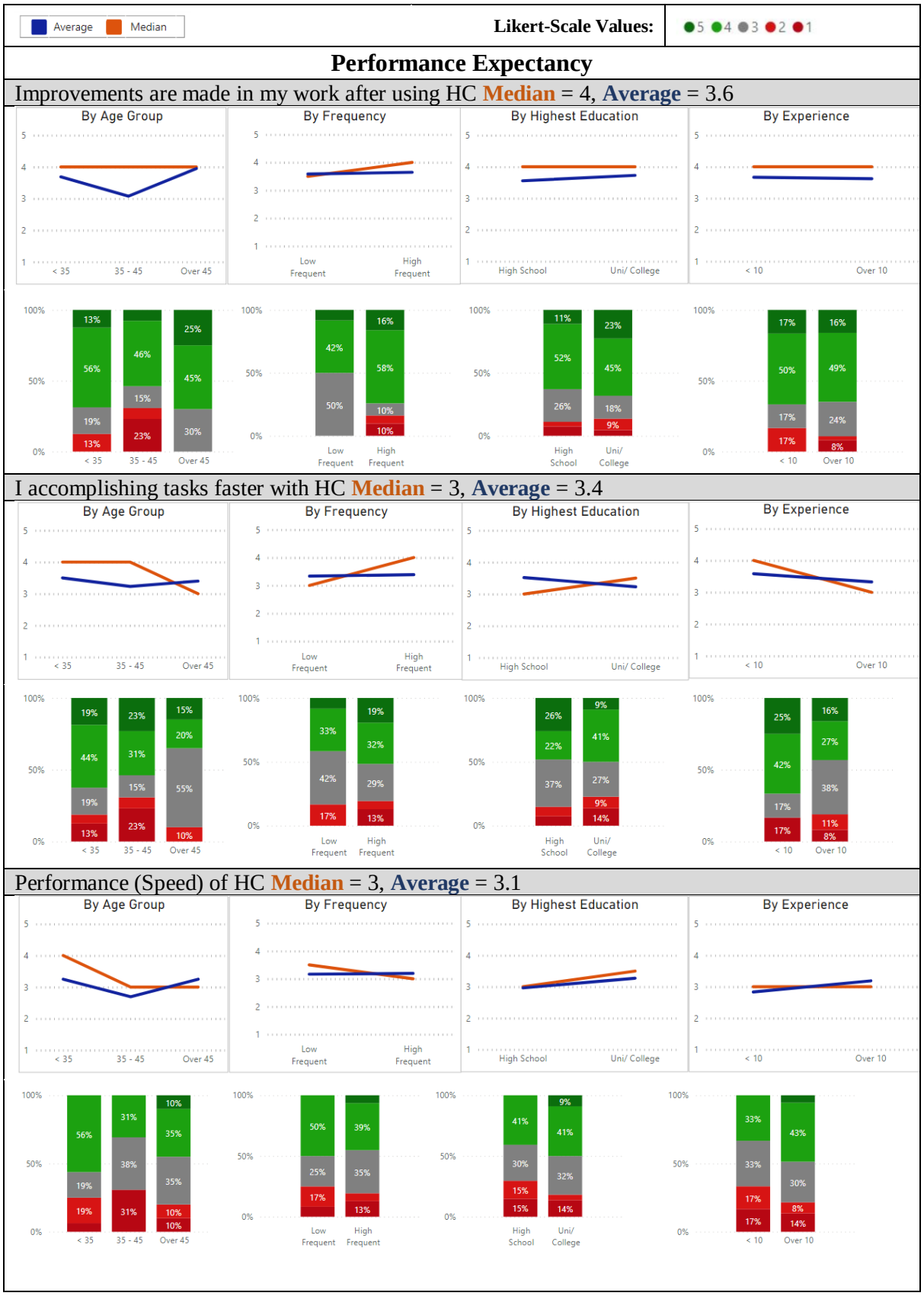

Rate HC
Strongly disagree
Neither agree nor disagree
Strongly agree
1
2
3
4
5
I feel improvements are made in my work **after** using HC
HC helps accomplish things **faster**
Learning how to use HC is **easy for me**
Rate HC
Strongly disagree
Neither agree nor disagree
Strongly agree
1
2
3
4
5
The **management** thinks that I should use HC
HC is useful for **my co-workers**
I have the **knowledge** necessary to use HC


Rate HC
Strongly disagree
Neither agree nor disagree
Strongly agree
1
2
3
4
5
Using HC is enjoyable
Using HC became a habit
I always try to use HC
I'm satisfied with HC performance (speed)
Frequency of using HC
Hourly
Daily
Weekly
Monthly
Others


Your company size (Number of employees)
1-50
51-100
Over 100
You use HC for (Optional)
Your email address to further contact you (Optional)
Top 3 most important features in HC (Optional)
Improvements suggested for HC (Optional)
→

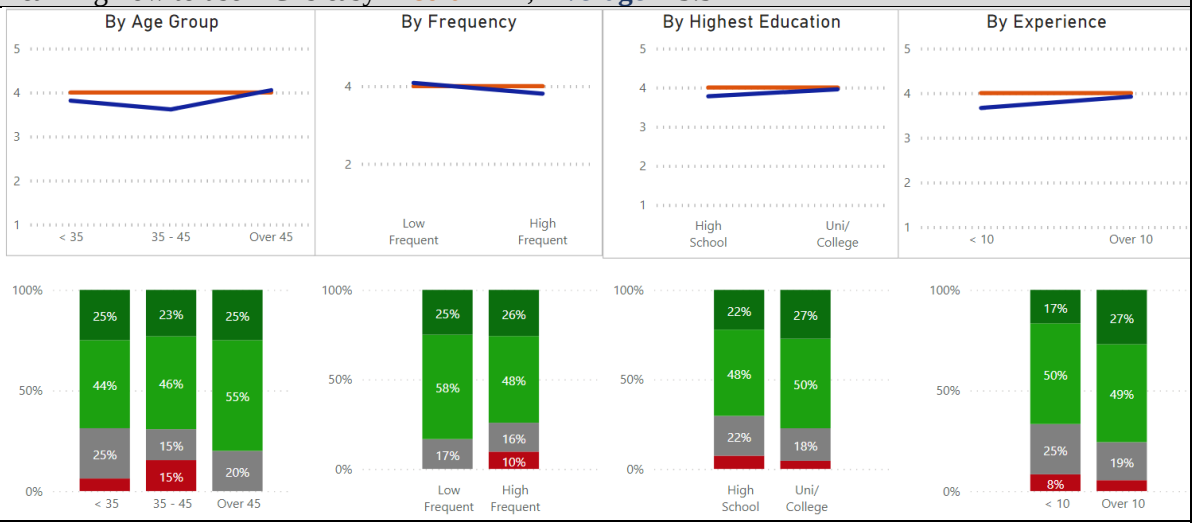
Appendix C: Survey Charts using UTAUT2 Constructs and Variables

Details of the survey results using the UTAUT model and the extension of it UTAUT2:



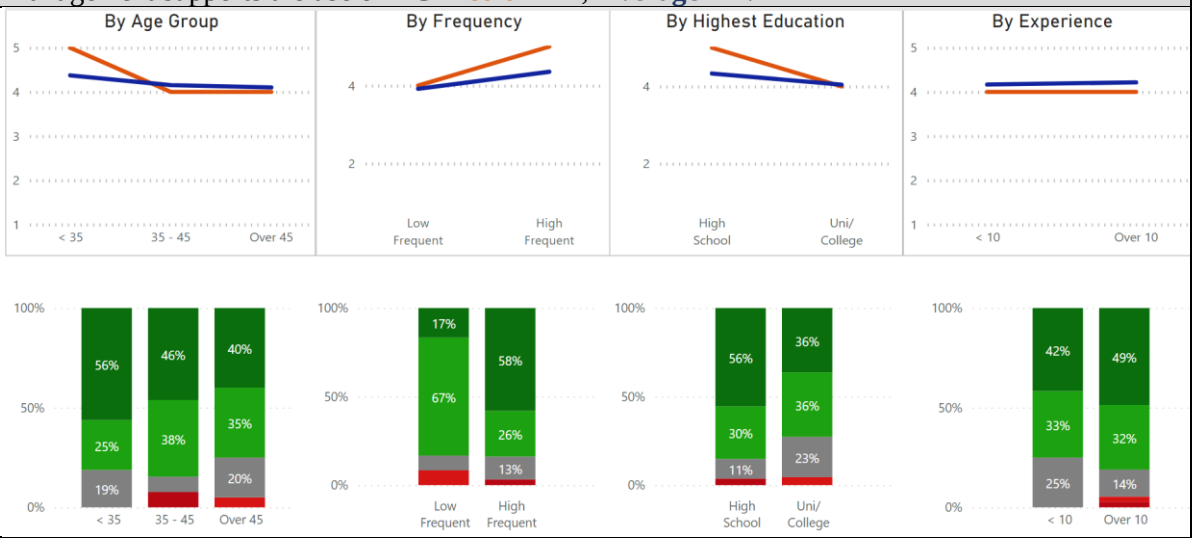
Effort Expectancy

Learning how to use HC is easy **Median** = 4, **Average** = 3.9

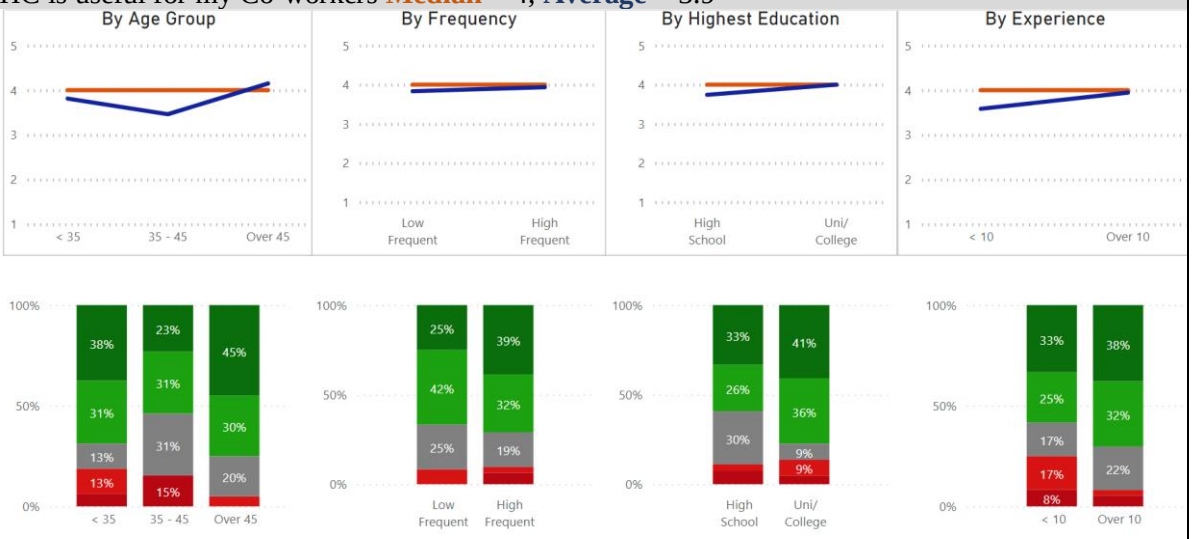


Social Influence

Management supports the use of HC **Median** = 4, **Average** = 4.2

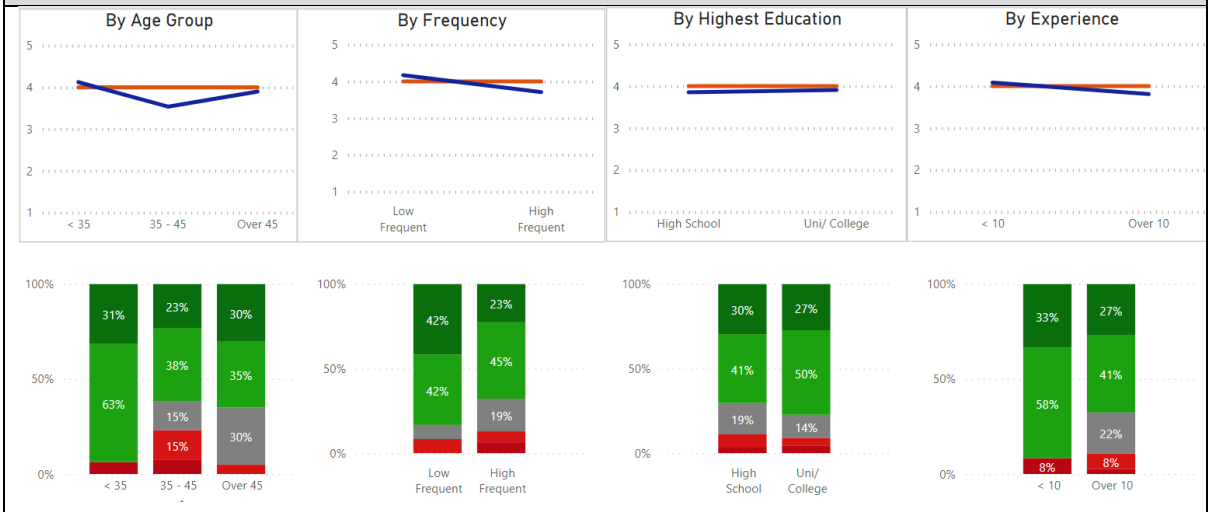


HC is useful for my Co-workers **Median** = 4, **Average** = 3.9



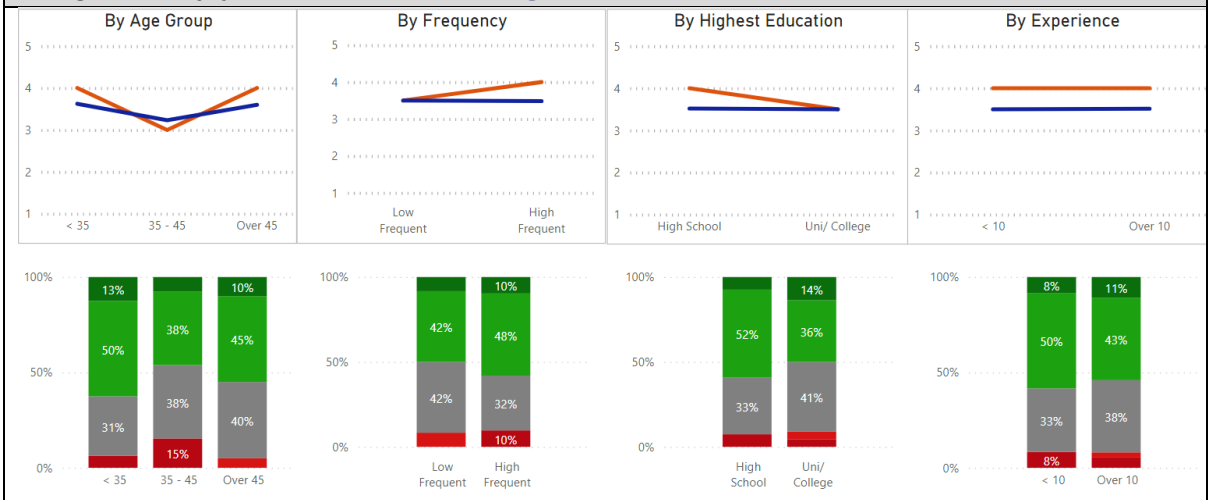
Facilitating Conditions

I have the available knowledge to use HC **Median** = 4, **Average** = 3.9



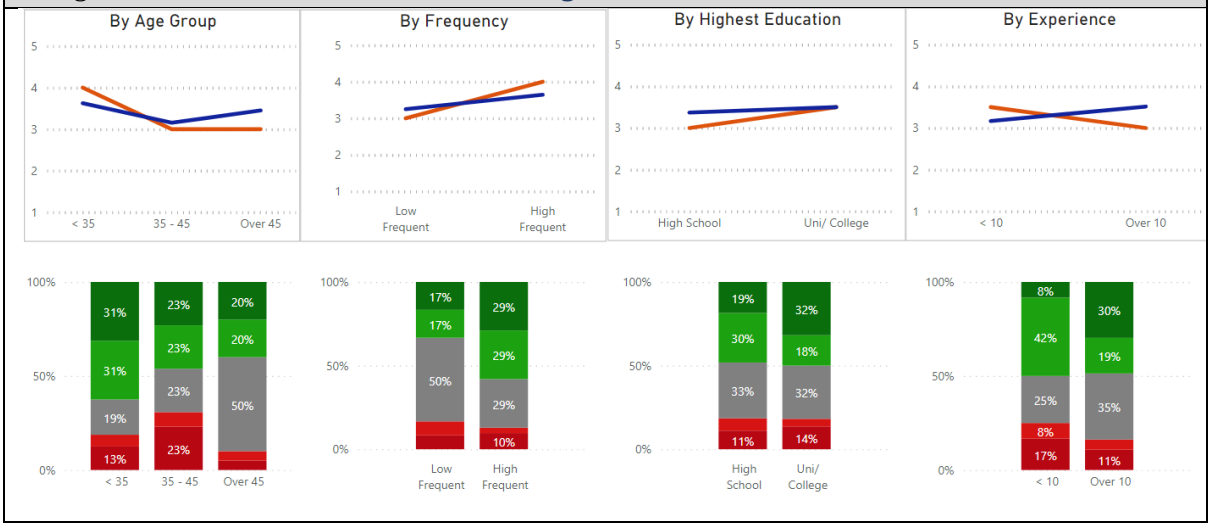
Hedonic Motivations

Using HC is enjoyable **Median** = 4, **Average** = 3.5



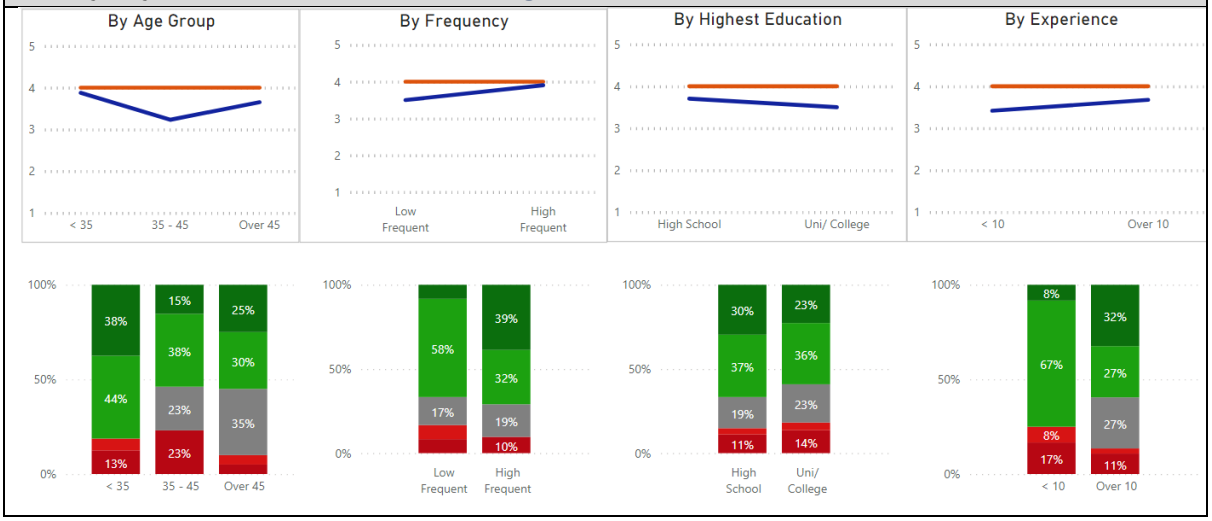
Habit

Using HC became a habit **Median** = 3, **Average** = 3.4



Behavioural Intention

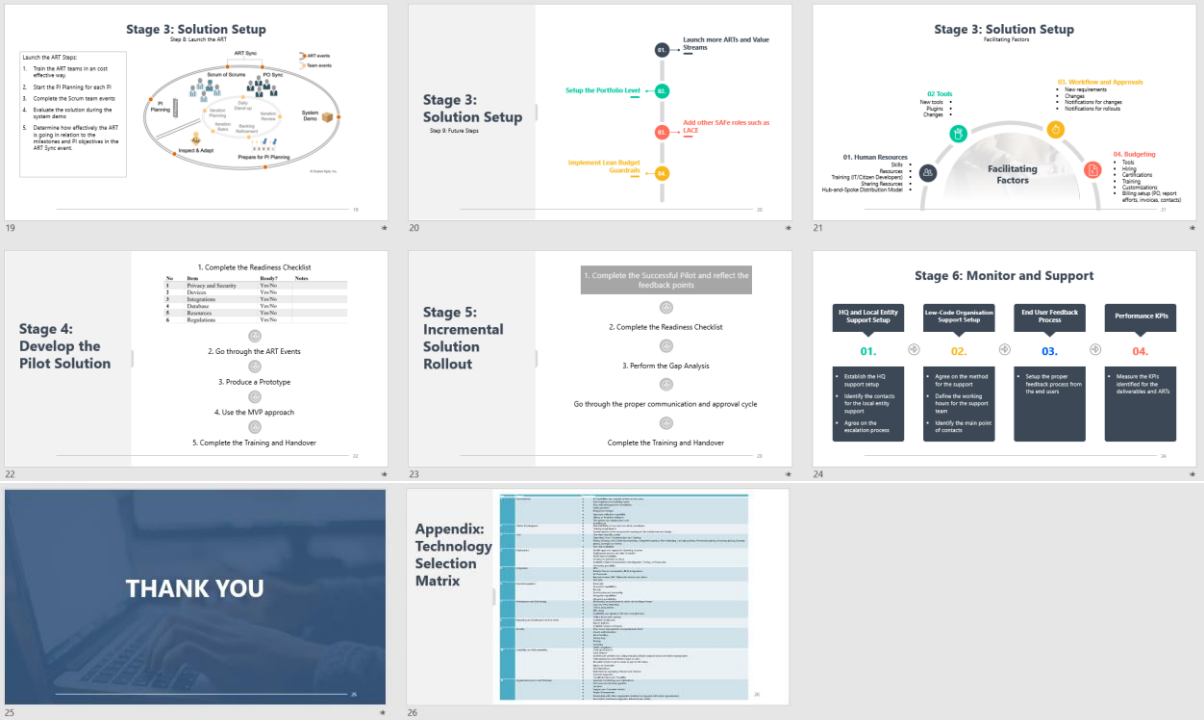
I always try to use HC **Median** = 4, **Average** = 3.6



Appendix D: Proposed Rollout Framework Design Presentation

The proposed rollout framework design was presented using the following slides:

A global rollout methodology for low-code/no-code platforms for a business with high local autonomy



Appendix E: Technology Evaluation Matrix

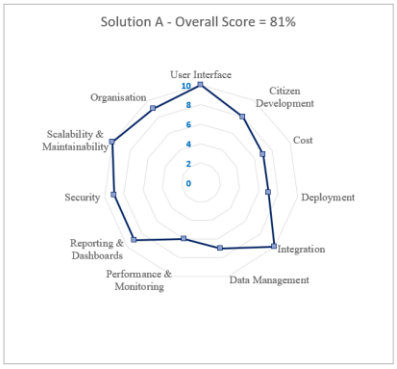
No.	Aspect	Description
1	**User Interface**	• UI Capabilities and controls offered to end users • User experience for building forms • Easy onboarding process for builders • Styles provided • Responsive designs • Signature collections capability • Library of Workflow elements • The options for scripting low code • Multilingual
2	**Citizen Development**	• Maintainability of use cases for citizen developers • Training requirements • System reports issues and provides warnings in the architecture and design
3	**Cost**	• One-time, monthly, yearly • Operating Costs, Customizations and Training • Pricing Strategy used (Value-based pricing, Competitive pricing, Price skimming, Cost-plus pricing, Penetration pricing, Economy pricing, Dynamic pricing strategies or others) • Free trial availability
4	**Deployment**	• Mobile apps and supported Operating Systems • Deployment process and time to market • White label availability • Hosting On-premise or Cloud • Available solution environments: Development, Testing, or Production • Versioning possibility
5	**Integration**	• APIs • Robotic Process Automation (RPA) integrations • IoT Protocols • Barcode readers, NFC, Bluetooth devices and others • GPS APIs
6	**Data Management**	• Data Lake • AI and ML capabilities • BI tools • Data hosting and ownership • Encryption capabilities • Migration possibilities
7	**Performance and Monitoring**	• Monitoring and performance charts for developed forms • Logs and error reporting • Trends and patterns • KPIs setup • Availability and uptime of the low-code/no-code platform • Offline Mode with caching
8	**Reporting and Dashboards for End Users**	• Available dashboards • Report builders • Available formats for Export

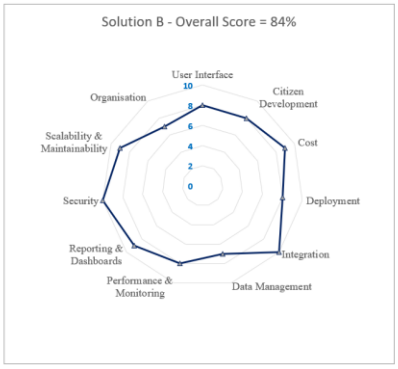
9	**Security**	• User access management and permissions level • Secure authentication • Error handling • History logs • Backup • Archiving • GDPR compliance
10	**Scalability and Maintainability**	• Scale up resources • Load balance • Architecture sufficient for scaling including Global, regional and local entities management • Onboarding new and different types of users • Reusable Content such as forms or part of the forms • Impact of downtime • Data limitations • Maintenance, managing changes and versions • Software upgrades • Troubleshooting and Testability
11	**Organisation (Low-code/no-code platform)**	• Expertise (Technology and Operations) • Use Cases and industry specifics • Location • Support and Customer Service • Project Management • Partnership with other organisation (isolated or engaged with other organisations) • Innovation (continuous upgrades, infrastructure, vision)

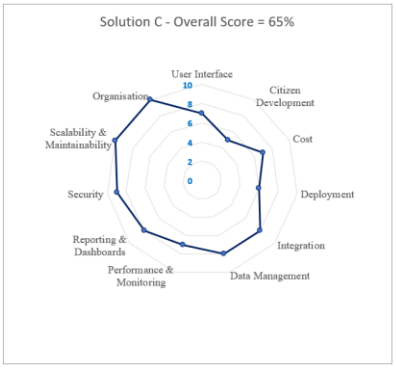
Each of the eleven aspects needs to have a weight that suits the organisation and a score assigned for the available solutions to decide on the shortlisted ones. The total weight for all aspects should be 100%, the Score is set as a value from zero to ten where ten indicates the highest score. To calculate the value for each aspect, this formula is used: **(Score/10)*Weight**.

Below figure illustrates a sample of the evaluation matrix and the results displayed in the web chart:

Aspect		Weight	Solution 1		Solution 2		Solution 3	
			Score	Value	Score	Value	Score	Value
1	User Interface	5%	10	5%	8	4%	7	4%
2	Citizen Development	25%	8	20%	8	20%	5	13%
3	Cost	15%	7	11%	9	14%	7	11%
4	Deployment	15%	7	11%	8	12%	6	9%
5	Integration	5%	10	5%	10	5%	8	4%
6	Data Management	5%	7	4%	7	4%	8	4%
7	Performance & Monitoring	5%	6	3%	8	4%	7	4%
8	Reporting & Dashboards	10%	9	9%	9	9%	8	8%
9	Security	5%	9	5%	10	5%	9	5%
10	Scalability & Maintainability	5%	10	5%	9	5%	10	5%
11	Organisation	5%	9	5%	7	4%	10	5%
	Total	**100%**		**81%**		**84%**		**65%**


Solution A - Overall Score = 81%
User Interface
Citizen Development
Cost
Deployment
Integration
Data Management
Performance & Monitoring
Reporting & Dashboards
Security
Scalability & Maintainability
Organisation
10
8
6
4
2
0


Solution B - Overall Score = 84%
User Interface
Citizen Development
Cost
Deployment
Integration
Data Management
Performance & Monitoring
Reporting & Dashboards
Security
Scalability & Maintainability
Organisation
10
8
6
4
2
0


Solution C - Overall Score = 65%
User Interface
Citizen Development
Cost
Deployment
Integration
Data Management
Performance & Monitoring
Reporting & Dashboards
Security
Scalability & Maintainability
Organisation
10
8
6
4
2
0