Karelia University of Applied Sciences
Bachelor of Engineering
Industrial Management

Analysing the Efficiency of the Software Development Process in a Low-Code Technology Company

A Case Study Using Value Stream Map

Katharina Abelita

Thesis, December 2023

www.karelia.fi

Karelia AMMATTIKORKEAKOULU

THESIS
December 2023
Degree Programme in Industrial Management

Tikkarinne 9
80200 JOENSUU
+358 13 260 600

Author
Katharina Abelita

Title
Analysing the Efficiency of the Software Development Process in a Low-Code Technology Company: A Case Study Using Value Stream Map

Commissioned by
Forward Forever Oy

Abstract
Low-code platforms address the rising need for technology solutions by streamlining development through drag-and-drop features. However, bottlenecks in other stages can still prolong the overall software development process. This thesis examines the low-code software development process primarily aiming to uncover inefficiencies and propose solutions to tackle them.

Two case studies revolving around Power Apps development were examined to gain insights into the process. Using literature review, thematic analysis of hour entries, and interview with the lead developers, a visual representation of the process was created through a value stream map.

Key findings unveiled substantial delays in both case studies, primarily stemming from the extended wait times for external parties' input or availability. Notably, granting access rights by clients emerged as a recurring challenge while internal practice relied on various interfaces for work planning, target setting, progress tracking, and update management. The multitude of communication platforms for project-related information exchange presented an additional hurdle. In response to these challenges, a set of potential solutions was proposed, encompassing streamlining project management, improving client communication, automating status updates, optimising meetings, and being cautious with holiday project initiations. Although these proposed solutions may offer process improvements, they require real-world implementation and continuous feedback for validation, while financial considerations warrant further investigation.

Language
English

Pages 50
Appendices 4
Pages of Appendices 5

Keywords
value stream map, low-code software development, efficiency, current state analysis, lean wastes

Contents

1 Introduction

One of the main reasons low-code platforms are utilised in software development is to respond to the ever-increasing demand for technology solution in a more agile manner. Through the platform's drag and drop features, the steps in the development process are reduced and makes it faster versus the traditional way of writing long lines of code. However, the entire process of software development also involves other stages other than designing and developing, such as gathering requirements, providing credentials to the developer and ideation of the product. If bottlenecks are experienced in one or more of these other stages, the entire software development process may still end up being lengthy.

The purpose of this research is to identify these bottlenecks or wastes in the entire software development process by mapping the current state using a value stream map. The company under study is an Information Technology consultancy firm focusing on the low-code/no-code tools of the Microsoft Power Platform suite. One of the services it provides to its customers is app development using Microsoft Power Apps. This study uses two case studies involving Microsoft Power Apps projects. Thematic analysis and semi-structured interviews were used to collect and verify data that were then analysed to provide solutions to eliminate or reduce the identified wastes. Afterwards, a future state value stream map was created reflecting the proposed solutions and their proper implementation.

2 Value stream mapping in software development

Value Stream Mapping (VSM) is a lean methodology which aims to map out the end-to-end steps in the development of a product and categorise activities as either (1) non-value-adding, (2) necessary but non-value adding, or (3) value-adding. It was made popular by Toyota when they applied it to their production system (Shingō, 1989) and has since been applied by many other manufacturers such as in steel production (see Adbulmalek & Rajgopal, 2007), precast concrete component factory (see Wang, Tang, Zou & Zhou, 2020) and casting industry (see Gunaki, Devaraj & Patil, 2021). A typical value stream map contains three major components namely *information flow*, *work flow*, and a *timeline* and among

these, the timeline is the biggest selling point of a value stream map as it has proven to be a great contributor in driving improvement to operations design (Martin & Osterling, 2014). For example, in a study by Singh, Garg, Sharma and Grewal (2010) where VSM was used to aid the lean implementation in a production industry, some reported improvements resulted to a reduction in work-in-process inventory (-89.47%), finished goods inventory (-17.85%), lead time (-83.14%), processing time (-12.62%), and required manpower (-30%) while output per operator increased by 42.86%.

As the tool proved itself to be helpful in improving efficiency and productivity, it has also been later adopted by the service sector, including software. One of the earliest papers mentioning the use of VSM in software development is that of Middleton & Joyce (2012) where it involved a nine-person team back in 2008 to draw the development life cycle stages onto kanban boards and all the work at each stage were recorded on cards that were attached to the boards. Immediately, the team was able to see that there were more work-in-progress and bottlenecks than they initially thought. Although this seems like a simple exercise to conduct, Anand, Chandrashekar and Narayanamurthy (2014) have noted that because VSM was originally developed for the manufacturing industry, there are differences in how the tool should be used in the software industry, particularly noting the predominance of information flow over material flow, absence of inventory, invisibility of certain processes, and role of customers, among others.

Khurum, Petersen and Gorschek (2014) describe the Value Stream Mapping Process in the context of software intensive product development to start with *Initiation* and followed by *Current process map*, *Waste identification* and *Process improvement*. They also suggest adding *Retrospective analysis* as the final step to allow clarification of possible misunderstandings and further improve the value stream process the next time it is conducted. The *Initiation* phase involves preparation and planning activities such as key stakeholder identification, purpose definition, team definition and training, problem identification and scoping, and value creation definition. The second phase, *Current process map*, is when the tasks and flow are identified, data is collected, and the current state

map is created. This is further explained in the succeeding section. The third step, *Waste identification*, goes beyond simply labelling certain processes as waste, but also identifying and documenting the reasons and root causes. In the *Process improvement* phase, strategies on how the wastes can be reduced or eliminated are suggested and its effects are documented in the future value stream map that is also created at this stage. (ibid.)

2.1 Creating a current state VSM

The current state value stream map is used as basis to identify wastes and the reasons for why they occur (Khurum et al., 2014). Martin & Osterling (2014) provided a template for a current state value stream map for service industries which was slightly revised in Figure 1 to have a meaningful colour scheme and to use icons available in Microsoft Visio which was used to create the maps in this research.

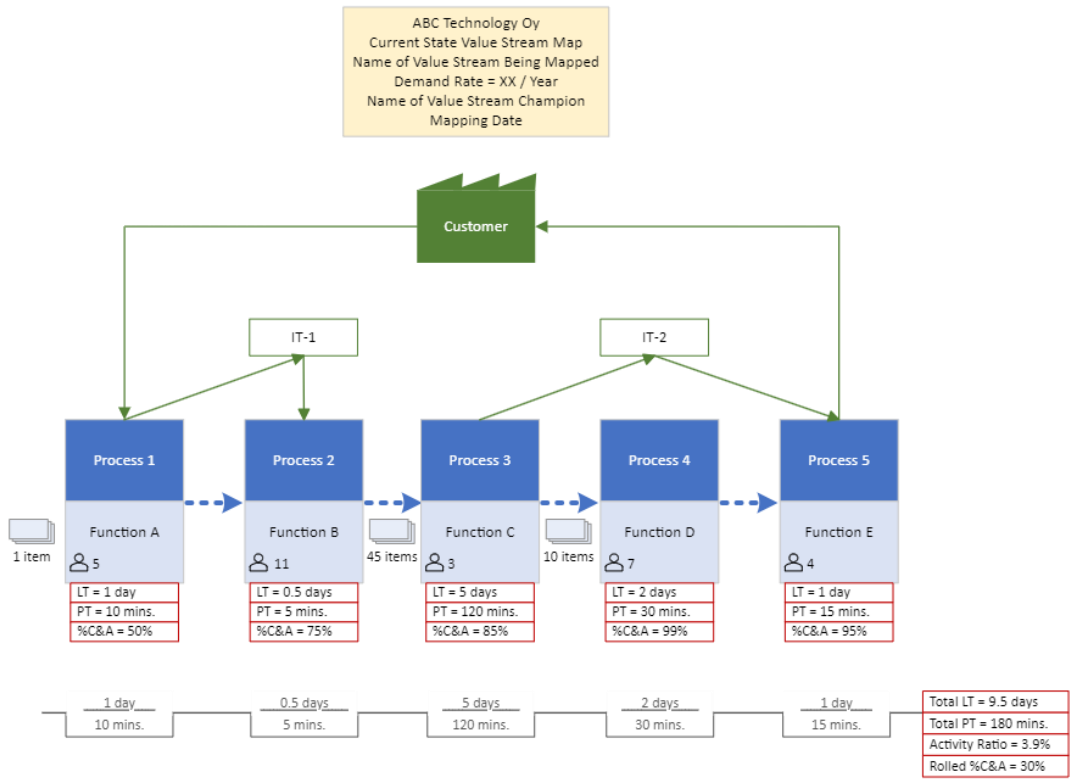


Figure 1. Current state value stream map template**.** Adapted from Value stream mapping: How to visualize work and align leadership for organizational transformation, by K. Martin & M. Osterling, 2014, Copyright 2014 by Karen Martin and Mike Osterling.

In Figure 1, the major components are represented by green for information flow, blue for workflow, and grey for timeline. The information flow can be interpreted from the map as starting from the customer (e.g. a list of requirements for an app) and is received by the first process block (Function A) as indicated by the arrowhead direction. The information is then entered to a system called IT-1 which then becomes the source of information for Function B. It is important to identify all systems and applications that each process interface with and also note whether or not there is communication between and among the systems. In this way, technology-related disconnects, voids, and redundancies can be easily visualised in the map (ibid.). As can be inferred from Figure 1, there is a technological disconnect between IT-1 and IT-2 systems.

The workflow component shows the amount of work-in-process between or within the process blocks. In the template, Function A has one (1) work item in queue, while Function C has 45 items and Function D has 10 items. Functions refer to the team/department/individual that perform the process indicated in the same process block. For example, Process 1 could be Requirements gathering and Function A could be the Deployment team. On the bottom-left corner of the Function box is the number of people who are currently performing the described activity. This is worth noting since it is helpful for work balancing or resource designation to certain value stream when designing the future state. Finally, the process blocks are connected by dashed push arrows, indicating that work is being pushed through the value stream. In cases where a parallel process occurs, they are still visualised as process blocks on the same vertical position as its parallel process. However, the timeline critical path or the path with the longest lead time still needs to be identified. The process blocks that are part of the timeline critical path are the ones whose lead times and process times brought down to the timeline (ibid.)

The timeline component is drawn from the performance metrics (in red boxes), particularly the lead time and process time of each process block. Lead time is the total time it takes from when a work is made available to a worker or team of workers (e.g. when a customer submits a change request) until it has been completed (e.g. requested change has been deployed to production). Process

Time is the total time used to complete a task (e.g. development work), including *talk time* necessary to obtain or clarify task-related information, *read and think time* for processes that undergo review or analysis, and even *nonhuman work time* involved to process work (e.g. file upload time) (Martin & Osterling, 2014). Lead time and process time are common performance metrics for current state evaluation. These metrics will serve as benchmark for the improvement efforts resulting from the current state analysis so the choice of performance metrics to show on the map depends primarily on its purpose. Another commonly used as a metric for most office and service value streams is percent complete and accurate which measures the quality of the output of each process. It is simply the percentage of the time that downstream customers receive work that can be used as it is without needing to make corrections to the provided information, supply lacking information that should have already been provided, or clarify information that shouldn't have been vague to begin with. This measure helps identify the root cause for poor quality (ibid.). Other metrics that can be used include total production cycle time, production total value added time, and total non-value-added production time. In the case of Gunaki et al. (2021), they used these as metrics on their value stream map which was used to aid in process optimization at a die casting company.

After all necessary information and metrics have been gathered, a summary metrics is displayed on the right-most side of the timeline. The summary metrics in Figure 1 includes the Total Lead Time (sum of all lead times), Total Process Time (sum of all process), Activity Ratio (Total PT/Total LT * 100) and Rolled %C&A (product of all %C&A * 100). An Activity Ratio of 3.9% can be interpreted as work being idle 96.1% of the time. Meanwhile, a Rolled %C&A of 30% means that 70% of work needs to be reworked at some point throughout the value stream (Martin & Osterling, 2014).

2.1.1 Identifying process blocks

The process of building software may vary within and among organizations, depending on different factors such as the past experience of the project or product manager, market sector of the organization carrying out the development work, and size and scale of the organization (Coleman & O'Connor, 2008).

Nevertheless, all programs undergo the same lifecycle – conception, requirements gathering/exploration/modelling, design, coding and debugging, testing, release, maintenance/software evolution, and retirement – with some programs possibly compressing certain steps or merging at least two steps into one set of work (Dooley, 2011). The same applies to software development using low code technology with steps simplified into ideation, analysing, planning, designing and developing, testing, publishing, and maintenance (Mäntyomena, 2020).

To optimize the development cycle, there are many different models which determine the chronological order of these steps. In general, these models fall under either the traditional plan-driven models type or the newer agile development models type (Dooley, 2011). Some of the most popular models include Waterfall, Iterative, Spiral, and Agile. Among these, Agile is said to be more efficient than the other models due to its iterative and incremental nature (AWS, n.d.), and can promote better communication among the development teams and team members (Karvonen, Behutiye, Oivo and Kuvaja, 2017).

For processes that adopt the principles and values of agile, software is built in increments from the start of the project instead of a one-time delivery at the end. One model which follows the Agile methods is the Scrum model, where it starts with a product backlog containing the list of features to be implemented during software development. Then it goes through sprint planning/meeting where the sprint duration is identified, and the scrum team chooses which user stories from the product backlog are the most important. The members also decide on how to turn the stories into a series of tasks that can be accomplished on time and who is assigned to which task. Then teams break out to work on the sprint. Daily scrum meetings commence to monitor the status of the tasks and each member reports on what they have done for the sprint goal. It is possible that after every sprint, there will be a shippable product increment to be shown to the customer. A sprint review is carried out at the end to analyse and discuss the overall results. A sprint retrospective is also done to identify how to improve the development process on the following steps. After points of improvement are identified, the team starts to focus on the next sprint planning. (Middleton & Joyce, 2012).

2.2 Wastes in software development

Once tasks that make up the process are identified, their value-add is assessed. In manufacturing, tasks are considered value-adding if it directly impacts the product (e.g. milling, welding, painting). Khurum et al. (2014) note that from a software product development perspective, evaluating value should be viewed from all angles, particularly *customer*, *internal business*, *innovation* and *financial*, since value tend to be more indirect. From the customer's point of view, the product or service's value is based on aspects such as its usefulness, inherent worth, functionality, reliability, user-friendliness, ease of maintenance, and adaptability. It also takes into account the value derived from the delivery process, network effects, supplementary value, user experience, the long-term value of customers, customer retention rates, as well as the financial impact on customers in terms of both expenses and revenue (Khurum Gorschek and Wilson, 2013).

Tasks that do not add value are then classified as either necessary yet non-value-adding or pure waste (also referred to as unnecessary non-value-adding). However, waste may mean differently for different people and in many companies, there is no clear definition of waste and not many are actively identifying or attempting to remove wastes in their organization (Alahyari, Gorschek and Berntsson Svensson, 2019). In manufacturing, the seven commonly accepted wastes as per the Toyota production system are *inventory*, *inappropriate processing*, *overproduction*, *transport*, *waiting*, *unnecessary motion*, and *defects* (Hines & Rich, 1997). In the context of agile software development organisations, these wastes can be translated into *partially done work*, *extra processes*, *extra features*, *task switching*, *waiting*, *motion* and *defects* (Poppendieck & Poppendieck, 2007).

Partially done work are those software projects that remain undeployed and possibly become obsolete. Unless these projects are deployed to production, they continue to tie up resources in investments and have yet to prove if they will solve the business problem. *Extra processes* mostly refer to paperwork that no one bothers to read. It is considered a waste if there is no one waiting for what is being produced by the process. *Extra features* are new code or technical capability that are not needed now. These are considered waste since it adds

complexity and becomes potential point of failure. *Task switching* occurs when a person is assigned to multiple projects at the same time. As they switch from one project to another, this becomes a waste since they take time to gather their thoughts and get into the new flow. *Waiting* is self-explanatory and in many cases, is caused by delays. *Motion* refers not only to the physical movement of a person (e.g. walking down the hall to find out the results of a test) but also to the movement of artifacts, particularly handoff, as a big amount of tacit knowledge do not get transferred from the creator to the receiver. *Defects* are also self-explanatory. They become a bigger waste the longer they remain undetected. In addition to these seven wastes, *management activities* are also worth noting to be potential waste. These activities are considered waste if they do not directly add value to the product yet greatly impacts waste in an organization. (Poppendieck & Poppendieck, 2007).

2.3 Process improvement and future state map

After the wastes have been identified and evaluated, the next step is to find ways to eliminate them and design the future state. Prioritising which wastes to eliminate is based on which category the work effort belongs to—value-adding, necessary but non-value-adding, or pure waste. Highest priority and attention should be given to eliminating pure wastes, followed by reducing the effort to complete necessary yet non-value-adding work, and finally, minimising the work effort put on value-adding tasks (Martin & Osterling, 2014). Khurum et al. (2014) provide a list of some possible strategies that can be taken to address certain wastes. For example, when tackling delays, possible strategies include implementing an average or pseudo takt time for work processes, coordinate time efficiently, ensure information availability through visual, physical flow, and pull-based approaches while eliminating bottlenecks in the workflow (Poppendieck and Poppendieck, 2008 as cited in Khurum et al., 2014).

Once there is a comprehensive overview of what will be done, the next step is to identify which functions will carry out the activity at each process, and estimate the new process time, lead time and %C&A. These estimates may need to be later adjusted as more relevant information is acquired. The projected summary metrics (i.e. Total Lead Time, Total Process Time, Activity Ratio, and Rolled %

C&A) are computed based on the projections and added to the Future State Value Stream Map as exhibited in Figure 2.

A future state value stream map reflects how the value stream could potentially look like and the activities that need to be done to get there. Aside from the improved timeline and metrics, other notable changes in the future state value stream map (Figure 2) as compared with the current state map (Figure 1) are less Process Blocks, IT-1 and IT-2 systems communicate with each other (lightning bolt arrow signifies automatic upload of data), and appearance of kaizen bursts (yellow star shapes). The improvement activity is shown on a map inside a kaizen burst and positioned to where it is applied. Upon completion of the future state VSM, it is recommended to have it with the key stakeholders and relevant managers to ensure there are no issues that may have been left out and agree on the future state design (Martin & Osterling, 2014).

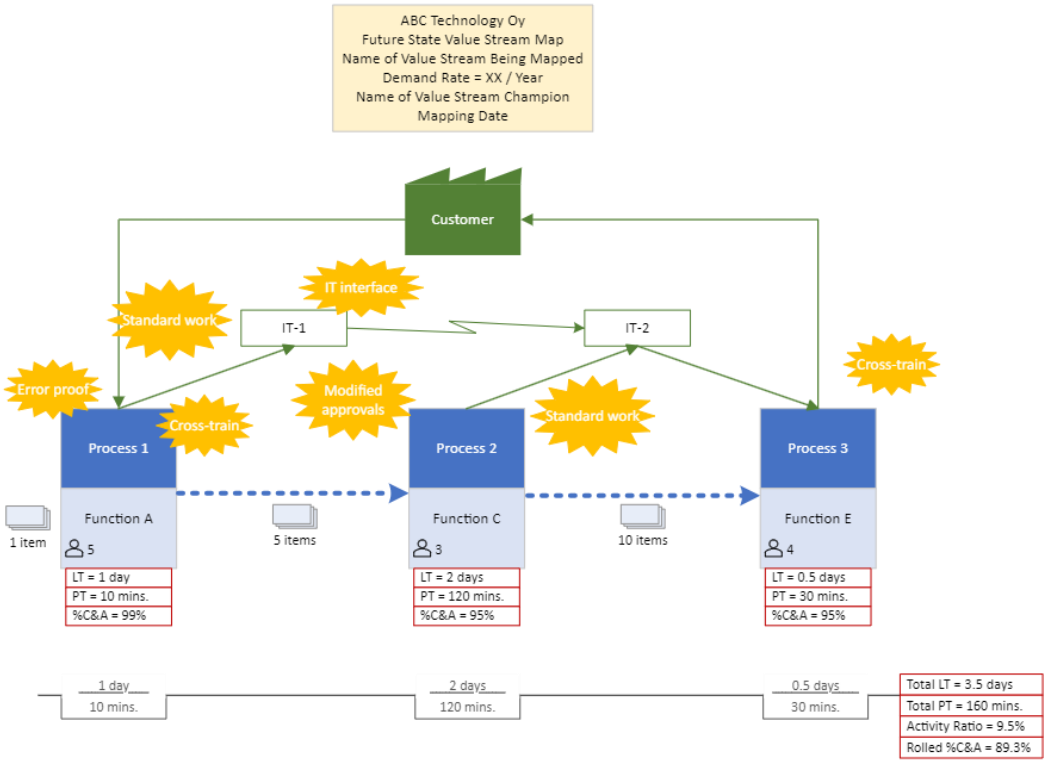


Figure 2. Future state value stream map template. Adapted from *Value stream mapping: How to visualize work and align leadership for organizational transformation*, by K. Martin & M. Osterling, 2014, Copyright 2014 by Karen Martin and Mike Osterling.

3 Methodological choices

To achieve the goal of identifying wastes in the low code software development process and create a plan for process improvement and waste elimination, this thesis used case study as the research approach. In particular, only two (2) customer projects in the focal company were studied. The data collection methods were literature review, thematic analysis of hours logged for the cases and a semi-structured interview. Qualitative methodology was chosen since the focus of this study is to explore complex and subjective phenomena such as the understanding of the experiences and perspectives of individuals involved in the software development process.

A literature review was carried out in order to develop a theoretical framework that guided the research and to gain better understanding of the research design, data collection, and analysis procedures (Creswell, 2014). In particular, existing research, theories, and practices in software development and value stream mapping were collected and reviewed. Afterwards, two (2) cases were selected from the focal company's internal Customer Relationship Management (CRM) system (also referred to as Business Forward) and hour entries logged by the employees for the cases were analysed using thematic analysis. In conducting a thematic analysis, the researcher was able to take a well-structured approach in processing the raw data (King, 2004 as cited in Nowell, Norris, White and Moules, 2017). The process involved data familiarization, creating preliminary codes, searching for theme connections, reviewing themes, and creating theme definitions and names (Nowell et al., 2017). The resulting themes were the process blocks needed as first step to create the current state value stream map. A process block column was added to the raw data and each hour entry was assigned the best-fit process block. As the tabular raw data also included the amount of hours it took the employees to do the actual work, the day it happened, and who did the work, the total number of workers, process time and lead time metrics were calculated through Power BI. The map was then visualized in Visio according to the template presented in Figure 1.

Once the first presentable version of the map was drafted, the main employees involved in the case were interviewed to clarify vagueness in the data and to verify

the correctness of the map (see structure of the meeting in Appendix 1). The map was then revised accordingly. In addition, the participants were asked regarding their experiences, opinions, and perceptions regarding the software development process. Interview questions were open-ended so that participants can give in-depth responses and allow the researcher to collect detailed and nuanced data from them (Denzin and Lincoln, 2011). These interviews were also recorded and transcribed for analysis.

This research was commissioned by the company under study. Research permission was not required from them as the research aims, methodology, and data collection methods were discussed with the company representative during the planning phase. Nonetheless, informed consent was still obtained from the participants before they were interviewed. The researcher provided the participants with a clear and detailed explanation of the research aims, methodology, and data collection methods, as well as of their rights as research participants. This included their right to withdraw from the study at any given point. They were then asked to sign a consent form indicating their willingness to participate in the research (see Appendix 2).

4 Current value stream map development process

The development process of this research composes of three (3) phases: (1) planning and preparation, (2) data collection, analysis and validation, and (3) solution development. The work breakdown structure for each phase is illustrated in

Figure 3 and further elaborated in the subsequent sections.

4.1 Planning and preparation

Provided that the researcher expressed her interest in process efficiency, the commissioning party suggested to study the time elapsed on each phase of the software development process. The hypothesis of the commissioning party is that since the company uses low-code technology, the bottlenecks occur not in the development phase but in the pre-development period which includes waiting time for the client to provide the necessary access rights for the developer to start working.

The researcher had then selected Value Stream Map as the tool of choice given that it has been taught in many of her classes and she already had some experience in developing and using this tool. Nevertheless, she felt the need to still conduct an extensive literature review on the topic since her knowledge and experience was limited to the application of the tool to the manufacturing scene and needed more understanding on how it can be applied to the service sector such as software development. Likewise, a literature review on the typical software development process was done in order to gain better understanding on the phases that needed to be mapped and analysed. Appropriate methodological choices and ethical considerations were also reviewed to ensure that the results of the study would be credible and reliable.

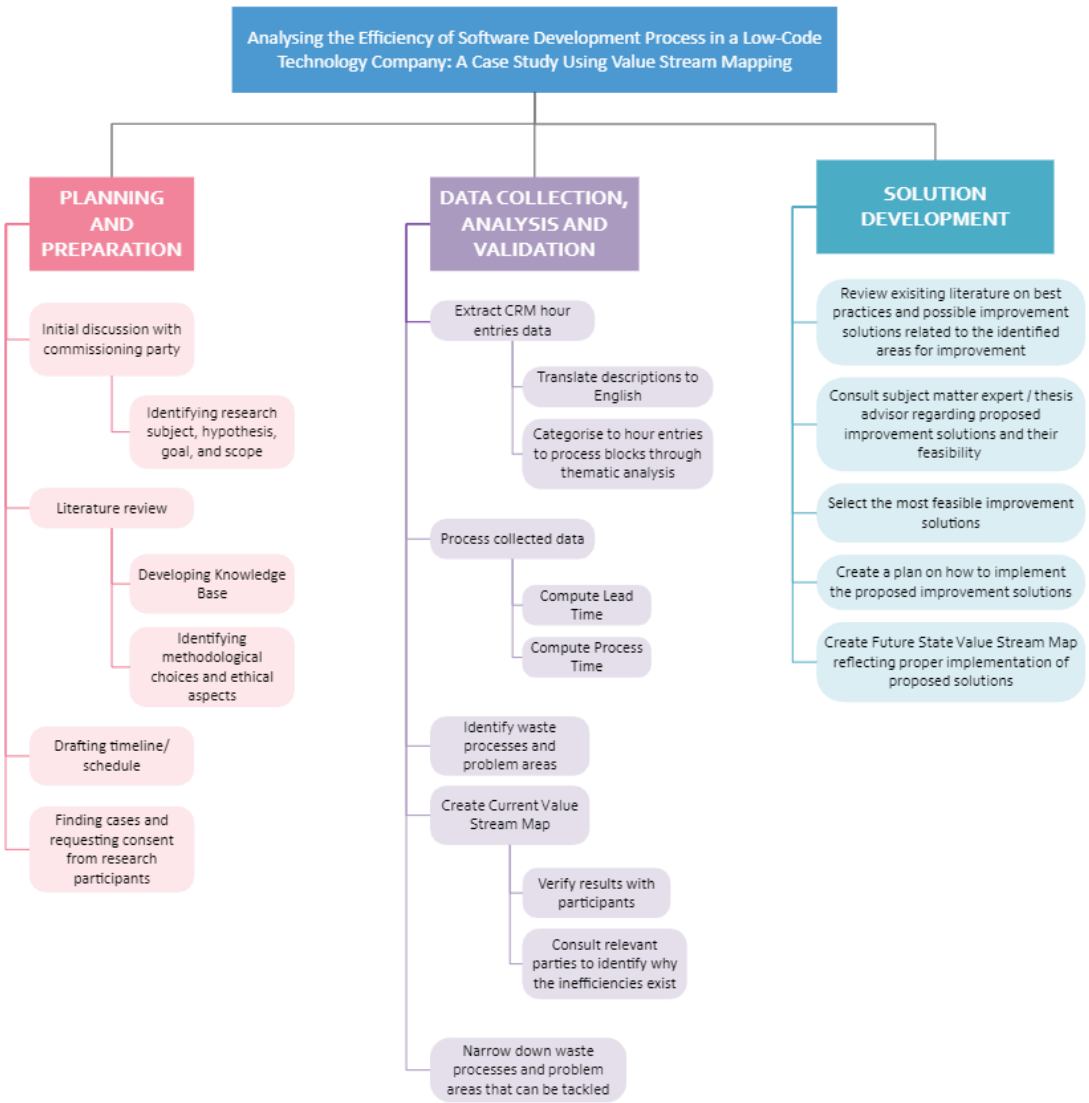


Figure 3. Revised work breakdown structure for the research.

After developing the theoretical framework to guide the research and formulating the research design, a schedule was made. To ensure that the thesis would progress as scheduled, a work breakdown structure was created and each task was given a due date and tracked using Microsoft Lists and Outlook Calendar. The tasks created were based on the initial research design which was focused on cases that were starting in August 2023. The initially planned data collection methods were literature review, and a combination of observation work on the cases and semi-structured interviews with the employees involved. In selecting cases to be studied, a company-wide “call for participants” was announced during the team meetings in June and the same was published in the company’s “Announcement” Teams channel asking for volunteer participants who have Power Apps cases starting in August 2023. Responses to the call were low and those that responded were either not ideal because they were not Power Apps projects or were still pending for approval from the customer.

By mid-August, the data collection method was revised since the researcher was targeting to finish her thesis by the end of the year and was already falling behind schedule. The new data collection method targeted cases that were already completed or were nearing completion and these cases were identified using the company’s internally developed Customer Relationship Management system called “Business Forward”. Through the system, the researcher was able to see all ongoing and completed cases (a.k.a. work orders). All employees would log their hours daily on their internal Hour Entry App and assign those hours to a work order and provide brief descriptions on what work has been done at that time. The Hour Entry App records are integrated to Business Forward which also stores other information about the customer account, including information shared on the dedicated Teams channel for that customer.

In selecting which customer accounts and work orders will be used for the research, the work orders were sorted according to which had the most recent hour entries. All non-Power Apps projects were excluded, and the resulting work orders were then skimmed through to check which had detailed hour entry descriptions, used English for hour entry descriptions and which ones had estimated total hours long enough to cover the pre-development phase as well.

Qualified work orders are of “time and material” contract type, meaning that the client compensates the contractor according to the amount of time spent by the contractor’s staff in completing the project. Thus, all activities and logged hours in the work order are value creating both from the perspective of the customer and internal business. The specific value followed in this research is the time to market, particularly from project initiation to customer acceptance/approval. Lead time, process time, and activity ratio were the main metrics analysed in this case. Meanwhile, %C&A and work-in-progress were not included since they cannot be inferred from the data and were also difficult to estimate since most of the work in different process blocks were done by the same individuals.

From all the qualified work orders, two cases were shortlisted, and the responsible developers were contacted to ask for their consent to be participants. They were also informed about the research goals and the new approach. Both participants agreed and the data collection phase begun with priority given first to the first case. After the current and future value stream maps for the first case was completed, then the same processes were applied to the second case.

4.2 Data collection, analysis, and verification

This thesis was designed to be qualitative in nature with case study as the research approach, particularly involving only 2 customer cases in the focal company. The data collection method was revised and changed to thematic analysis since the source of data were Excel file tables containing the hour entry records of the employees for the work orders associated with the selected client organisations. The extracted Excel file, a snapshot of which is shown in Table 1, contains a table of the WorkDay (when the work was done), Hours (number of hours spent for the work), WorkDoneByUser (who logged the hours and did the work), Description (description of the work done), WorkOrder (the project associated with the specific hour entry), Account (the name of the client organisation), and Created On (the date and time the hour entry was entered into the system).

Work Day	Hours	WorkDone ByUser	Description	WorkOrder	Account	Created On
26/04 /2022	1.00	Txxx	xxxxx projektin hyväksyminen	xxxxxx Avustushakemusp rosessi toteutus	Sxxxxxxx	16/05/2022 7.30
14/04 /2022	2.00	Txxx	xxxxx technical introductions video recording	xxxxxx Avustushakemusp rosessi toteutus	Sxxxxxxx	19/04/2022 4.56
13/04 /2022	0.50	Oxxx	xxxxx muutospyyntöjen selvitykset	xxxxxx Avustushakemusp rosessi toteutus	Sxxxxxxx	19/04/2022 6.15

Table 1. Snapshot of raw data.

4.2.1 Background of the cases

The first case involves a religious organisation in Finland (also referred in this research as Case 1 organisation, client, customer) which provides assistance to families and individuals who face unexpected financial crises or emergency situations. In the year 2021, the organisation's aid fund processed a total of 612 grant applications with 525 receiving a positive decision. This amounted to €772,380 in grants distributed through various channels. The focal company was tasked to modernise the application process using Dataverse for Teams. The objective was to streamline and enhance the grant application process, reduce manual work, improve monitoring and reporting, bolster security, and ensure robust auditing capabilities. A total of five (5) individuals (also referred in this research as developers, advisors, consultants) from Forward Forever (also referred in this research as the commissioning party, focal company, company under study, contractor) worked on the project from the Proof of Concept (POC) phase to the production launch and client acceptance.

Meanwhile, the second case focuses on a Finnish company (also referred in this research as Case 2 organisation, client, customer) specialising in the manufacture and service of industrial equipment. This client had to change their Power Apps subcontractor due to recent global events and had then reached out to Forward Forever to continue the Power Apps development. The app will be mainly used for risk evaluation and was originally utilising SharePoint for data storage, and Canvas app for the user interface. There is one consultant from Forward Forever who is mainly in-charge of the development for this case and from time-to-time was supported by two other colleagues. Some major tasks were

to migrate the data to Dataverse, refactor codes, complete the remaining backlog tasks, and deploy the app. Contrary to the first case, this does not involve a POC phase as the app was already built. As of 01 October 2023, the case was still ongoing although deployment tasks have already begun. In the interest of time, the activities covered by this research only involves the hours logged up until 01 October 2023 when the hour entries were downloaded and analysed.

4.2.2 Workflow

4.2.2.1 Case 1 process blocks

Case 1 organisation had multiple work orders associated with it as the account has been active for more than two (2) years now. As of 18 August 2023, there were a total of 559 hour entries recorded under this account. The scope of the research was then defined to only include work orders with the keyword *avustushakemusprosessi* (translating to “grant application process”). This included three (3) work orders which covered the POC phase, app development, launch to production, and project approval. The excluded work orders were for maintenance work and change requests that occurred after the approval. After filtering out the work orders, a total of 303 hour entry records were shown and upon examining the data, it was discovered that the earlier hour entries had descriptions in Finnish. Thus, these descriptions were first translated to English using Google Translate and added as a new column to the Excel table. In addition, the researcher took note of the announcement when the proposal was formally approved by the client which was posted on the dedicated Teams channel and added it as an hour entry to the Excel table.

The data analysis started with a thematic analysis of the Excel file. The first step was to get familiarised with the data. Workday was sorted to be ascending to better understand how the project progressed from day 1. Afterwards, columns were added to classify the data into categories. The first column is a Yes/Blank column to identify if the customer was involved in that particular hour entry. Whenever unfamiliar names or the word “client” were mentioned in the description (e.g., meeting with Person A or demo with client), the column was marked “Yes”. If the description somewhat signals a meeting but does not specify

with whom (e.g., Azure meeting), the column was marked “Yes?” to verify later if it was indeed with the client or not. Otherwise, the column was left blank if the description did not indicate the involvement of a third party. There were repeating hour entries that indicated an involvement of an organization, 2NS, which was not the client. Upon looking it up online, it was found that it was an information security testing and auditing company. This was also categorized as “Yes?” to later verify their role in the project.

For the Process blocks, these were first identified on two (2) levels—Phase and Work item (see Table 2). The Phase column was to identify the stage of the hour entry and the Work item column had the uniform codes on what work was done in that phase. The work item column would be used as basis to identify value-adding and non-value adding tasks. The choices for both columns were initially based on the case’s Project Plan schedule. For the Phase column, the choices were revised to have a more linear structure since the Project Plan schedule clearly showed that many Phases overlap, and this would cause problems in the value stream map. Thus, the typical software development process stages taken from literature was also integrated to the choice list.

Similar to the Phases choice list, the Work item codes were also initially based on the case’s project plan. However, there were hour entries that did not fit with the pre-defined choice options, and so new choice items were created for those. After the initial round of coding of Phases and Work items, the data was analysed using Power BI. The hours were aggregated monthly according to the phases and were then visualised as a Gantt chart (see Appendix 3). The visualisation made it apparent that the phases overlapped one another on many instances and made it difficult to be mapped. Thus, a revision of the codes was needed to have a clear sequence from one phase to another.

Phases	Work items
- Requirements gathering	- Planning/Preparation
- Proof of concept	- Credentials
- Technical specification	- Specification / architecture
- Design/ Architecture	- Design and development
- Data modelling	- Audit/Compliance
- Designing and developing	- Review
- PBI Report development	- Meeting
- Archiving workflows	- Demo

- Integrations - Training - Usability test & pilot use - Production launch - Documentation - Approval	- Environment configurations - Form app (Power apps) implementation - Testing and debugging - Process for archiving form and attachments in Azure Data Lake - Case Management System + Payment Process - Training - Support for pilot users, small development work based on feedback - Communication support and other necessary launch measures - Documentation - Finalisation - Design / Architecture - Auditing - Specification - Approval

Table 2. Initial codes for hour entries.

The new approach involved adding a new column to the Excel file called “Process blocks”. The choice items for this column were developed based on the visual timeline, combinations made between Phase and Work order, as well as the original hour entry description. The goal was to have more specific codes that would make the process more sequential. The initial run produced 25 process blocks codes (see Appendix 4), which were again visualised as a Gantt chart in Power BI and mapped using Visio.

The sequence of the processes was made according to the Power BI Gantt chart. There were still multiple Process Blocks which had its start and end days coincide with each other and were then visualised in the map as parallel Process Blocks stacked vertically. When the VSM’s timeline was being created, it became apparent that the Process Blocks needed to be revised again since having parallel processes meant that the process times and lead times for only one Process Block can be carried down to the timeline. By selecting only one process block, the process time for the other parallel process blocks will be ignored even though they are also part of the same timeline critical path and had the same workers involved (i.e., “Production architecture planning, design, and documentation” and “Business logic, UI/UX planning and design”). To resolve this, the overlapping timeline critical process blocks were merged into one so the process time would be more accurate. The final iteration of codes resulted to 18 Process blocks and are defined in **Error! Reference source not found.**.

Sort	Process block	Process block description
01	Initiation meeting	Initiation meeting
02	Account test and planning	Pre-POC Account test and planning
03	Kick-off workshop	POC kick-off workshop
04	POC development	Development of the proof of concept to showcase the feasibility of the solution
05	POC Demo	Includes the POC demo at board meeting, review walkthroughs before board meeting, Power BI report processing, and POC fine-tuning
06	POC Architecture review	POC Architecture review was done together with Production Architecture planning so the hours were split equally between the two
07	Production architecture, logic, UI/UX planning, design, and documentation	Planning, designing, and documentation for the production architecture, business logic, and UI/UX
08A	2NS / System review	Includes workshop and documentation work with 2NS and client
08B	Open specification meeting	Open specification meeting with client
09	Design, specification document review and revision	Design, specification document review and revision done after meeting with 2NS and client
10	Customer Approval, Kick-off meeting with client	Official approval by the client's board to carry out the project and consequent kick-off meeting
11A	Building the app, data model, workflow logic, testing, debugging	Activities related to sprint planning, application design and development, Azure configuration, internal meetings, workflow development, and Azure integration
11B	Report development	Report development work on the Power BI side.
12A	Launch to production, app modifications, & documentation	Deploy the app to the target environment, communication support and other necessary launch measures
12B	Webinar	Webinar with end users
13	Training / consulting	Training / consulting with client
14	Implementing change requests & documentation	Implementing change requests & documentation with end users
15	Project Approval	Client approves the project and work order is closed

Table 3. Case 1 final process block codes and description sorted chronologically.

4.2.2.2 Case 2 process blocks

Most of the processes applied in Case 1 were replicated for the second case. In the Excel file containing all hour entries for the Case 2 organization, the Work Order was filtered to only show those with the keywords “Risk evaluation App”. It resulted to 141 rows of hour entries which had descriptions in English so there was no need to translate them. The researcher then proceeded with thematic analysis by first getting familiarised with the data, followed by the identification of whether the customer was involved in the hour entry. Then an initial iteration of process block codes was done. Since majority of

the work done centred on app development, the initial codes were numbered as App development 1, 2, 3… and a new process block was created whenever the app development was interrupted by customer involvement (e.g. “Discuss with client: Flows”). This resulted to 32 process block codes, with App development from 1 to 11 and another 11 codes for status call and client discussions. As these were too many, the second iteration combined status calls and subsequent app developments. Processes that involved defining needs or specifications were kept as separate process blocks. This iteration resulted to 13 process blocks and was slightly revised after conducting the interview with the lead developer who highlighted that one process block was mostly a waiting period for the customer to sort out some internal processes (now named as “Customer-driven use cases definition with parallel app enhancements” process block) that was delaying progress in app development. The process blocks and corresponding descriptions are enumerated in

Table 4.

Sort	Process block	Process block description
01	Existing app evaluation	Walkthrough of the existing app and evaluating and analysing it.
02	Access rights & environments	Gathering requirements related to user, credentials, accesses, documentation, and environments.
03	Building the app, data model, workflow logic	Developing and refining various aspects of Canvas Apps, including refactoring, bug fixing, and implementing new features; Discussions and work on approval processes, conditional responses, and logic for Business Line models.
04	Demo with client	Demo of the updated solution to the client.
05	App modifications	Bug fixes and UI enhancements.
06	Discuss with client: Next developments and backlog	Call to align next developments and backlog.
07	Customer-driven use cases definition with parallel app enhancements	A period when the project was essentially on-hold as developers wait for the customer to define use cases. Activities that were logged were related to various modifications and fixes for data models, attachments, PDF templates, and app functionality and discussions to address issues and make changes based on testing results and user requirements.
08	Draft deployment plan	Drafting of deployment plan.
09	Continue development work 1	Integrating cloud flows with apps, making adjustments based on user feedback, planning integrations, implementing auto-save functionality, aligning status integration, and modifying app forms and integration flows to meet specific requirements and field mappings.
10	Call with client IT: Deploy to Prod preparations	Call with client IT: Deploy to Prod preparations.
11	Continue development work 2	Activities related to integration work, finalizing response functionality, updating deployment

		plans, and conducting status calls with the Business and IT teams to coordinate and review progress.
12	Launch to prod and app modifications	Tasks related to deployment, communication regarding API fields, mapping, testing, and finalizing Admin role functionality within the Canvas App for improved integration and functionality.
13	Testing session	Involves preparation and execution of testing sessions with test users, addressing email duplication issues to approvers, and holding key user discussions to outline tasks and next steps, followed by a status update call.

Table 4. Case 2 final process block codes and description sorted chronologically.

4.2.3 Timeline

Metrics were also calculated and analysed in Power BI. Pseudo-process blocks were created for idle periods in between two process blocks to account for them in the total lead time calculations. Lead time is the number of days elapsed from the first hour entry until the last hour entry belonging to the same process block. For lead times resulting to zero (0) days due to the first and last hour entries being the same, the total number of hours were computed and divided by 7.5 hrs which is equivalent to one (1) workday in Finland. Meanwhile, process time is the sum of hours logged in the hour entries belonging to that process block. Finally, activity ratio is the ratio between process time in hours and lead time in hours. To highlight the most inefficient process blocks (excluding the pseudo-process blocks), conditional formatting was also applied to the activity ratios as seen in **Error! Reference source not found.** and **Error! Reference source not found.**. The activity ratio is highlighted in dark red if it is less than or equal to 5%, red if it is more than 5% but less than or equal to 10%, and light red if it is more than 10% but less than or equal to 15%.

4.2.3.1 Case 1 timeline

Computing the actual elapsed time from the first process block—initiation meeting which occurred on 17/12/2020—to the final process block—project approval which occurred on 26/04/2022—the total is 495 days. However, in the initial timeline, the total lead time for the time critical path was 509.8 days. The difference is explained by the overlapping start and end dates between some of the process blocks (e.g. “building the app,…” and “launch to production,…”). To

minimise the error, the lead time measure for each process block was adjusted. The start and end days were not counted as full days but instead were computed as if the start day hours were done towards the end of the day and the end day hours were done at the beginning of the day. For example, the process block “Production architecture, logic, UI/UX planning, design, and documentation” started on 09/08/2021 with three (3) hours logged on that day and ended on 09/09/2021 with two (2) hours of work. The start day then amounted to 0.4 days and end day to 0.27. Adding these together with full days from 10/08/2021 to 08/09/2021, the process block lead time resulted to 30.67 days as seen in **Error! Reference source not found.**. With this adjustment, the total lead time was down to 503.97 days.

Summary of Process Blocks

Process Blocks	No. of workers	Customer involved	Lead time (days)	Lead time (hrs)	Process time (hrs)	Activitity ratio	Start	End
01-Initiation meeting	1	Yes	0.07	0.50	0.50	100.00 %	17/12/2020	17/12/2020
02-Account test and planning	1	Yes	0.07	0.50	0.50	100.00 %	18/12/2020	18/12/2020
02A-Idle	1		20.00	150.00			18/12/2020	08/01/2021
03-Kick-off workshop	3	Yes	0.93	7.00	7.00	100.00 %	08/01/2021	08/01/2021
04-POC development	3	Yes	26.07	195.50	45.00	23.02 %	08/01/2021	03/02/2021
05-POC Demo	2	Yes	0.67	5.00	5.00	100.00 %	04/02/2021	04/02/2021
06-POC Architecture review	3	Yes	34.40	258.00	12.00	4.65 %	15/02/2021	22/03/2021
06A-Idle2	1		139.00	1,042.50			22/03/2021	09/08/2021
07-Production architecture, logic, UI/UX planning, design, and documentation	3	Yes	30.67	230.00	59.75	25.98 %	09/08/2021	09/09/2021
08A-2NS / System review	2	Yes	33.67	252.50	34.50	13.66 %	07/09/2021	11/10/2021
09-Design, specification document review and revision	2	Yes	29.27	219.50	22.00	10.02 %	04/10/2021	03/11/2021
10-Customer Approval, Kick-off meeting with client	2	Yes	0.43	3.25	3.25	100.00 %	04/11/2021	05/11/2021
11A-Building the app, data model, workflow logic, testing, debugging	3	Yes	105.27	789.50	207.25	26.25 %	08/11/2021	22/02/2022
12A-Launch to production, app modifications, & documentation	3	Yes	60.40	453.00	113.00	24.94 %	14/01/2022	16/03/2022
13-Training / consulting	1	Yes	0.40	3.00	3.00	100.00 %	16/03/2022	17/03/2022
14-Implementing change requests & documentation	2	Yes	22.53	169.00	26.75	15.83 %	22/03/2022	14/04/2022
15-Project Approval	1	Yes	0.13	1.00	1.00	100.00 %	26/04/2022	26/04/2022
Total	**6**	**Yes**	**503.97**	**3,779.75**	**540.50**	**14.30 %**	**17/12/2020**	**26/04/2022**

Figure 4. Case 1 process blocks metrics analysed through Power BI.

The Activity Ratio for each process block was then calculated by first converting lead time from days to hours where1 day equals 7.5 hours. Afterwards, Process Time was divided against it and resulted to an overall activity ratio of 14.30%. **Error! Reference source not found.** shows the metrics for each process block with the last row showing the total. Meetings that constituted as their own process block resulted to an activity ratio of 100%. Excluding these meetings, the process block with the highest activity ratio was the development work itself (Process block 11A — Building the app, data model, workflow logic, testing, debugging). Meanwhile, the process block that had the lowest activity ratio was the POC architecture review with only 4.65% activity (excluding idle times—process blocks

2A and 6A). Worth noting, if idle times were included to the lead time of the process block prior to it, Account test and planning (Process block 02) would have had the lowest Activity Ratio of only 0.33%.

4.2.3.2 Case 2 timeline

Using the same method done in case 1, the total lead time for case 2 resulted to 354.93 days. However, the actual span of the project from the first process block (Existing app evaluation) to the final one (Testing session) was 360 days. The slight difference is explained by the adjustments in lead time calculation that was done in Case 1 which had overlapping processes. Unlike Case 1, the process blocks in Case 2 are flatter and do not overlap so it was not necessary to make the lead time adjustment but was still kept in order to have uniform calculations in both cases and taking into consideration that the difference between the actual lead time is not significant (5.07 days difference).

Summary of Process Blocks

Process Blocks	No. of workers	Customer involved	Lead time (days)	Lead time (hrs)	Process time (hrs)	Activitity ratio	Start	End
01-Existing app evaluation	2	Yes	2.27	17.00	4.00	23.53 %	03/10/2022	06/10/2022
01A-Idle1	1		1.00	7.50			06/10/2022	07/10/2022
02-Access rights & environments	2	Yes	5.13	38.50	1.50	3.90 %	07/10/2022	13/10/2022
02A-Idle2	1		4.00	30.00			13/10/2022	17/10/2022
03-Building the app, data model, workflow logic	3	Yes	55.87	419.00	116.50	27.80 %	17/10/2022	12/12/2022
03A-Idle3	1		1.00	7.50			12/12/2022	13/12/2022
04-Demo with client	1	Yes	0.13	1.00	1.00	100.00 %	13/12/2022	13/12/2022
04A-Idle4	1		3.00	22.50			13/12/2022	16/12/2022
05-App modifications	2	Yes	6.20	46.50	3.00	6.45 %	16/12/2022	23/12/2022
05A-Idle5	1		5.00	37.50			23/12/2022	28/12/2022
06-Discuss with client: Next developments and backlog	1	Yes	0.07	0.50	0.50	100.00 %	28/12/2022	28/12/2022
06A-Idle6	1		7.00	52.50			28/12/2022	04/01/2023
07-Customer-driven use cases definition with parallel minor app enhancements	3	Yes	189.40	1,420.50	48.25	3.40 %	04/01/2023	13/07/2023
07A-Idle7	1		6.00	45.00			13/07/2023	19/07/2023
08-Draft deployment plan	1		0.20	1.50	1.50	100.00 %	19/07/2023	19/07/2023
08A-Idle8	1		9.00	67.50			19/07/2023	28/07/2023
09-Continue development work 1	3	Yes	13.33	100.00	9.75	9.75 %	28/07/2023	11/08/2023
09A-Idle9	1		3.00	22.50			11/08/2023	14/08/2023
10-Call with client IT: Deploy to Prod preparations	1	Yes	0.13	1.00	1.00	100.00 %	11/08/2023	11/08/2023
11-Continue development work 2	2	Yes	6.40	48.00	4.00	8.33 %	14/08/2023	21/08/2023
12-Launch to prod and app modifications	1	Yes	23.40	175.50	21.50	12.25 %	21/08/2023	14/09/2023
12A-Idle10	1		1.00	7.50			14/09/2023	15/09/2023
13-Testing session	1	Yes	12.40	93.00	4.75	5.11 %	15/09/2023	28/09/2023
Total	**4**	**Yes**	**354.93**	**2,662.00**	**217.25**	**8.16 %**	**03/10/2022**	**28/09/2023**

Figure 5. Case 2 process blocks metrics analysed through Power BI.

Meanwhile, the total process time resulted to 217.25 hours and the activity ratio was only 8.16%. In the Power BI visualisation (see **Error! Reference source not found.**) which was used for analysis, the process block with the highest activity ratio (excluding the meetings that were also process blocks) was the initial development work “Building the app, data model, workflow logic”, similar to the first case. Those with the lowest activity ratios were the seventh process block “Customer-driven use cases definition with parallel app enhancements” with

3.40% and closely followed by the second process block “Access rights & environments” with 3.90%.

4.2.4 Information Flow

During the thematic analysis for both cases, it was also evaluated whether the customer was involved during the particular hour entry. On many occasions, it was explicitly mentioned that the meeting was done with the client while for those that were unclear, further investigation was made. Other places that were investigated to confirm the information flow were Business Forward and OneNote. Business Forward has a Timeline section where some email threads related to the case are visible. Emails are an obvious form of exchange of information and so they were translated from Finnish to English to understand the context. Dates of when the emails were sent and received were noted to associate them with a process block.

4.2.4.1 Case 1 information flow

OneNote served as the data repository for information related to case 1. All notes were in Finnish so they were machine translated to understand the context. The information stored in OneNote include the tasks that need to be done and who is responsible for it, meeting notes, and budget. The to-do list indicated that project management and tracking was done in OneNote and checkboxes next to the task were ticked when they were done. The meeting notes included the list of attendees and the date of the meeting which confirmed which hour entries and process blocks had customer involvement. There were also notes with Excel sheet screenshots signifying that data was also stored in Excel and was shared in OneNote. Lastly, a computation of the total hours spent on the project and the respective monetary equivalent was recorded in OneNote. From these, the IT systems and information flow was drawn on the map and later verified during the interview with the advisor-in-charge.

4.2.4.2 Case 2 information flow

In contrast to the first case, the second case had fewer associated documents such as proposals, schedules, and to-do lists. This is attributed to the second case's unique circumstances, starting with an existing app and bypassing the need for a Proof of Concept (POC). Likewise, there was no formal proposal

presentation requirement due to the absence of a competitive evaluation process. In Business Forward, it was noted that the initial contact originated from the client, who had learned about the focal company through a colleague already engaged with the company.

Client engagement was prominent throughout this case, with status meetings and discussions occurring at various process stages. The IT systems employed closely resembled those used in the first case, including Business Forward, Hour Entry App, Microsoft Power Platform, and Excel. One of the differences is that the second case shows Weekly Meeting App which started to be in use in the focal company after the first case ended. In the Weekly Meeting App, the researcher looked at the focus actions entered by the lead developer and for every focus action that referred to the case, an information flow line to Weekly Meeting App was drawn from process blocks which coincided with the focus action due date. Another difference is the project management tool used. In this case, the main tool used was on the client's side Teams channel dedicated for the project that was already existing even before Forward Forever took over the case. In the channel, there were Excel and PowerPoint files that contained the tasks for the project. For change request management, a SharePoint list embedded to the Teams channel was used.

4.2.5 Current state value stream map

4.2.5.1 Case 1 current state value stream map

After collecting, processing, and analysing the data related to the workflow, timeline, and information flow, the current state value map was completed and shown in **Error! Reference source not found.**. The project was being co-managed by two (2) individuals throughout the process. One of the co-leads was the main contact person of the client and had been involved in the project all throughout the process. The second co-lead was in-charge of app development (a.k.a. lead developer).

As seen in **Error! Reference source not found.**, value creation began with a brief initiation meeting with the lead developer where the customer described their needs. At this stage, billing, budget, and other important customer information were also entered into Business Forward. In addition, a work order was created

in the Hour Entry App and all hours related to the project were entered every workday. Thus, this and all other process blocks had information flow pointing to the Hour Entry App since the hours were logged every time work was done. Business Forward and Hour Entry App were both developed using the Microsoft Power Platform suite and thus have the same database allowing seamless integration of data between them (signified by the lightning bolt connection). Meeting notes were also recorded in OneNote which is not integrated to any other system used. Project tasks were also recorded and managed in OneNote.

Immediately after the initiation meeting was the account test and planning which was a prerequisite for creating a proof of concept on the client's environment. At this stage, the consultant informs the client of the necessary accounts/ licenses related to Office365/MS Power Platform which they will then set up for the consultant. Although the map shows that this process block had a lead time and process time of only 0.5 hours, it is should be noted that there was a 20-day delay from this process to the next one. Delays may have been caused by the holiday season as it took place in mid-December to early January of the following year.

Afterwards, a kick-off workshop was conducted together with the client to further discuss the needs and analyse the requirements and create a plan on the way forward. Worth noting, this process involved three (3) individuals from the focal company which caused the tripling of the process time although the kick-off workshop only lasted two (2) hours. The process time also included the preparation time for the workshop and thus resulted to seven (7) hours. The development of the POC immediately started on the same day as the workshop and work was mainly done using MS Power Platform. The process time took 45 hours while the lead time was about 27 days and a status check was conducted with the client during the POC development phase. The POC was then presented to the decision makers in the client organisation who evaluated whether to pursue the project with the consultant. An architecture review of the POC and other design and architectural studies were conducted in late March 2021, and an offer was sent to the client. After this, no work for the project was recorded for 139 days. It was found that the delay was due to the unavailability of the information security auditor (2NS) prior to the summer holidays. Through email

correspondence, it was agreed that a workshop with 2NS will be held in August and the consultant will prepare for it.

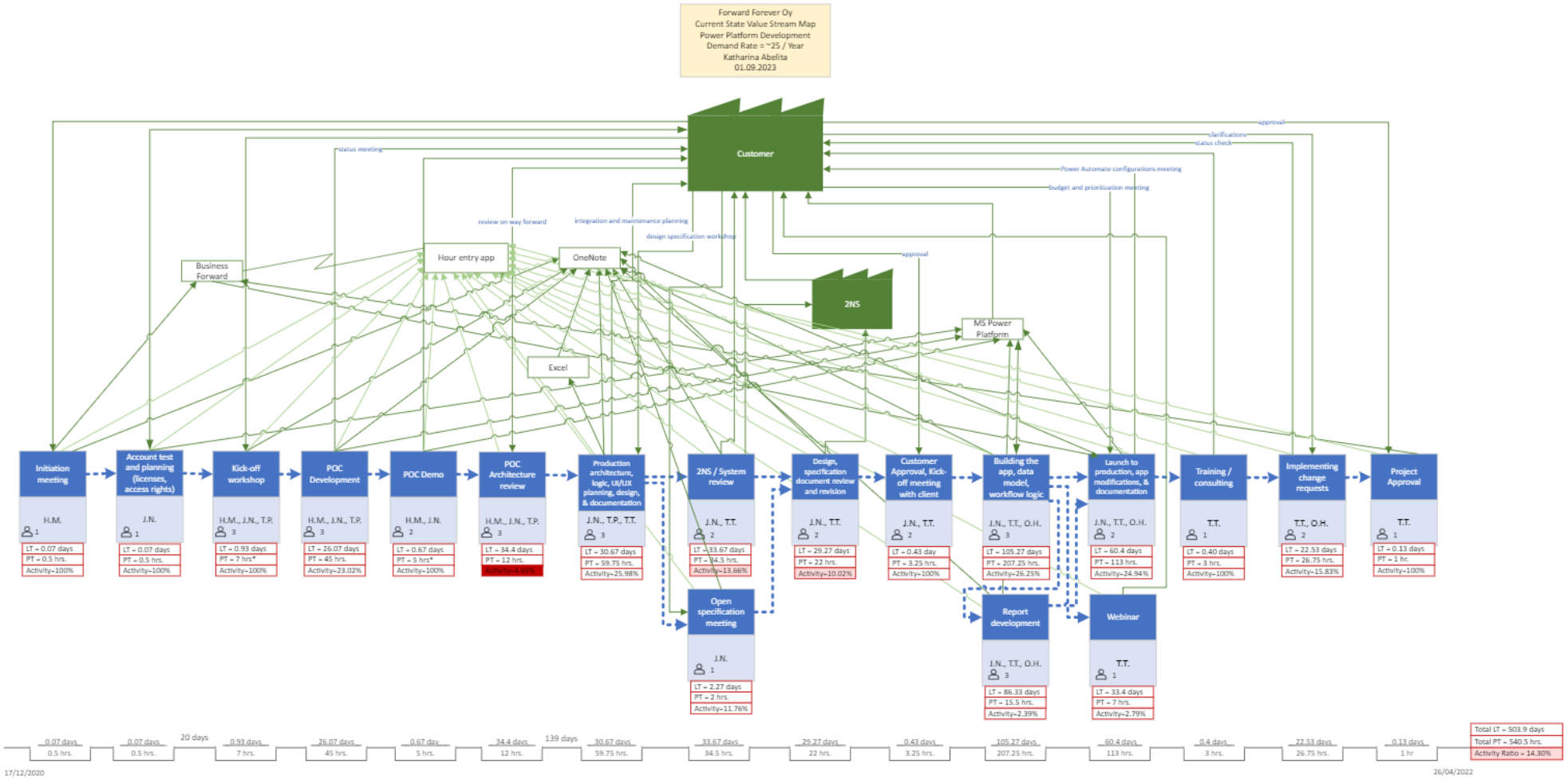


Figure 6. Case 1 current state value stream map.

Work resumed mid-August with a new lead developer taking charge of the project as the previous lead developer decided not to be involved in the project anymore. After the initial round of production architecture planning, design & documentation phase which spanned for about 31 days, the workshop with 2NS was conducted with the client also present. Before the workshop ended, an open specification meeting was done with the client signifying some changes would need to be done to the original plan. After revisions were made and documents were submitted to 2NS, a formal approval from the client was received confirming that the consultant will carry out the project.

Immediately, a kick-off meeting with the client was conducted and the development work for the app began. The lead time for this phase took about 106 days with process time of about 207.25 hours. According to the lead developer, at that time he was still not very experienced with Dataverse for Teams which was being used so he felt that the progress was slow in the beginning. Nonetheless, the activity ratio was maintained above 15% from this and all succeeding timeline critical process blocks. Likewise, a new worker joined the project at this stage which contributed to the improved activity ratio although some onboarding tasks needed to be done at first. As the app was being built, the report was also developed through Power BI which is also part of the MS Power Platform suite. It has been created as a separate block since it is not part of the timeline critical path (i.e., the value stream focuses on Power Apps development).

When the hour entry with the description “Azure subscription to Prod” was first recorded, it signified that preparations for production launch have begun. Although the app was already being prepared to be deployed, there were still modifications being made. Likewise, a budget and prioritisation meeting with the client was conducted during this phase and the relevant information for the meeting, particularly the total hours entered in the hour entry app were pulled from Business Forward and computations of the actual cost were made visible in OneNote. Prior to completing the launch, a webinar was also conducted to showcase the app to the users. After the launch, some training and consultation work was also done and was followed by some more change requests. After the

change requests were implemented, the project was accepted by the client and the work order has been also updated as completed in the CRM.

4.2.5.2 Case 2 current state value stream map

As pictured in **Error! Reference source not found.**, the value creation process commenced with the pre-development phase starting with an introduction to the existing app which was demoed to the advisors by the developer who built the app. The purpose was to handover the app development work and so it was also analysed and evaluated by the advisors and took about 3 days to do. Simultaneously, essential customer and project details, were recorded in Business Forward and a work order was created at this stage. The role of the Hour Entry App in this case is similar to how it was already described in the first case. After the app evaluation, requirements related to accessing the app in the customer's environment were gathered. Notably, the activity ratio for this process is one of the lowest in the entire process with only 3.9%. According to the lead developer, this is quite common for big companies who require an approval process before accesses are granted to external users. In addition, as part of the security model of the client company, access issues also propped up when the system detected that login attempts were done outside of Finland where the lead developer was located which has contributed to the low activity in this process block.

After the necessary accesses were received and issues were resolved, development work started. Tasks included refactoring of the Canvas App, making adjustments to remove SharePoint references, defining the Power Automate Flows structure needed for the new approval process, implementing logic for Business Line models, discussing AAD Groups, user access, and licenses with IT, and transitioning functionalities from Excel to Canvas App. SharePoint and Excel were then shown as IT systems used in this process block. This process spanned for about 56 days with actual work done for 116.5 hours. This was followed by a demo with the client showing the progress made, allowing the client to give feedback. Modifications were then made but worth noting, development work after the demo slowed down.

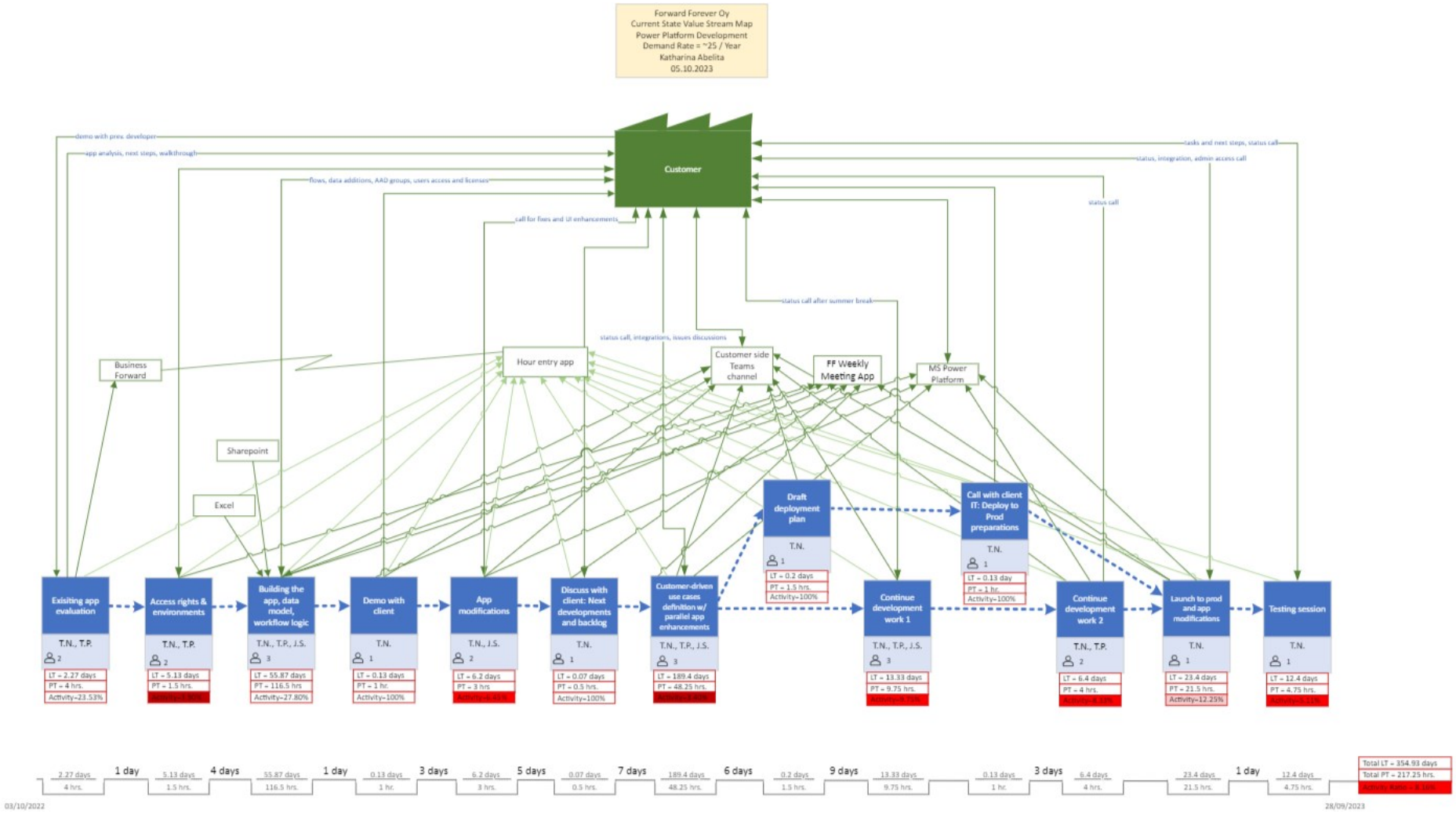


Figure 7. Case 2 current state value stream map.

As the end of 2022 was approaching, a discussion with the client regarding the next steps was done and work on the app resumed on 4 January 2023. According to the lead developer, the project was essentially on-hold at this point as they were waiting for certain specifications from another unit on the customer side. Nonetheless, some app enhancements were done during this waiting period and thus, did not reflect a long idle time which the lead developer recalled to be around 6 months. This included activities involving various modifications and fixes for data models, attachments, PDF templates, app functionality, testing, status calls, and user feedback. Work continued even during the summer months although the activity was at its lowest at 3.4% only.

Although development work continues, a deployment plan was also drafted quite early on, taking into account the slow response time from the client's IT department. As this task was related to deployment, it had a path separate from the continuation of development work and continued to the deployment preparation call with the client IT team. Nevertheless, this path as well as the development work are both timeline-critical, that is why the process blocks are not stacked vertically. The paths merged when tasks started to become more deployment-related in the "launch to production and app modifications" process block. The activities involved initiating deployment plan tasks, APIs and integrations, migrations, and fixing issues that propped up along the way. Finally, the app was handed over for user testing and the last activity logged in this work order as of 1 October 2023 was a status update call. The project continues to be active as of the date of writing this thesis, but the scope of this research is only up until what was done as of 1 October 2023.

4.2.6 Wastes in the value stream

The current value stream map was presented to and verified with each lead developer during a meeting where the Current VSM were presented to and verified with them and followed by a semi-structured interview. Key findings were found regarding delays caused by the customer and other external parties, technical challenges, client collaboration and project management, underscoring the complexities and opportunities encountered during the development process.

4.2.6.1 Case 1 wastes in the value stream

With a lead time of about 504 days, only 540.5 hours (or about 73 days) was the recorded process time. Idle time was mainly attributable to holidays and late availability of external parties. After the POC architecture review, the original plan estimated 38 to 42 working days to complete the project, yet the actual working days spent was around 67. Estimating work hours at the project's outset proved challenging due to changing requirements and unexpected technical issues. Technical issues are quite common in customer cases and in this instance, the lead developer recalls the issue regarding security group configurations which inadvertently affected data visibility for specific individuals. Meanwhile, the evolving nature of requirements was mostly due to the customer's uncertainty about their needs in the defining phase and lack of experience with IT. These resulted to difficulties in providing clear specifications and understanding the technical aspects of the project. Consequently, when new features were being requested during the development phase, it necessitated renegotiations of the budget or the removal of certain features to align with budget constraints.

The interview also revealed that it was the lead developer's first involvement in a large-scale development utilizing Dataverse for Teams. This platform presented differences in data modelling and data security compared to Dataverse, which he was more accustomed to. As a result, an initial learning curve was required, potentially contributing to slower progress in the project's early stage. Notably, the customer did not question the hour entries logged by the lead developer and all other individuals who worked on the project. They recognized the value of the work and were willing to compensate accordingly. This can be attributed to the high quality of output provided to the client, particularly with the user interface and user experience (UI/UX). The developer appreciated the creative freedom in designing the project's UI and UX with the client providing only the basic guidelines and not overly prescriptive about other details.

In terms of project management tools used, the lead developer finds OneNote to suffice for project management and tracking in two-person scenarios, but its effectiveness diminishes with larger teams. The developer personally resorted to using Excel for task monitoring, although he acknowledged uncertainty regarding

its suitability. He also noted that Microsoft Planner was employed in other customer projects, though it received mixed feedback. Identifying the most appropriate tool for project tracking, particularly for timely client updates, remained a challenge. Additionally, the current practice in the company requires upcoming plans and project status to be entered into other systems on top of OneNote or other personally preferred project monitoring tool. This process involves (1) entering monthly work allocations in the Hour Entry App, (2) creating weekly Focus Actions in the Weekly Meeting App, (3) entering completed work in the Hour Entry App daily, (4) marking the Focus Action as either completed or not completed at the end of the week, and (5) checking remaining budget in Business Forward. The developer expressed a preference for a unified data entry solution to streamline this workflow.

Finally, the low activity ratio was investigated by looking at the hour entries logged by the lead developer during the same period as the project was being done. It was found that the lead developer was also working on 20 other customer projects during the duration of Case 1. Other projects accounted for 48% of his customer hours and Case 1 accounted for 52%. On top of this, he also had hours logged for internal activities. Upon further investigation, it was found that the number of employees in 2021 was only 10 and increased to 14 in 2022 and at the time of this writing, the total is 22. Meanwhile the number of customers the company has worked with amounted to 71, 71, and 61 and the corresponding work orders totalled to 142, 189, and 216 for 2021, 2022, and at the time of this writing, respectively. These translate to a lowering in both Work Order-Advisor ratio from 14,2% in 2021 to 9.8% presently and Customer-Advisor ratio from 7.1% in 2021 to 2.7% currently. With less Work Order-Advisor ratio, it means that the advisor can allocate more time to the specific Work Order they are working on and do not have to resort to task switching as often as before.

4.2.6.2 Case 2 wastes in the value stream

The value creation process in the second case has various inefficiencies and waste, similar to the first case. This case also experienced extended lead times, with most process blocks having low activity ratios, resulting in an overall activity ratio of only 8.16%. These delays are primarily caused by waiting periods,

particularly during phases related to acquiring access rights and defining use cases.

According to the advisor-in-charge, acquiring access rights and environments from larger companies is a complex and time-consuming process due to the client company's IT procedures. Users are required to navigate to a specific portal, raise a ticket, and wait for the service team's response, which typically takes about two days to receive new user credentials. He also mentioned challenges in implementing multi-factor authentication due to his location outside Finland, which was immediately blocked by the existing IT security rules. This led to additional communication and back-and-forth interactions to finalise access to necessary environments, licenses, and passwords. The interviewee also emphasized that IT support's response time is generally slow, often requiring intervention from higher-ups to attend to his requests.

Furthermore, the project's progress is hindered by the delayed finalisation of requirements and use cases, primarily due to multiple business units involved in the app's use. These crucial elements are only completed after significant development work has been done, resulting in a project holdup for nearly half a year. This delay is further exacerbated during the summer months when many stakeholders are on vacation. The advisor highlighted that this was the biggest bottleneck in the project. Task switching was not the issue in this case but the slow pace that was decided by the client which meant that there were weeks with no progress made.

Challenges related to the development work itself included uncertainties about whether best practices were done in developing the app. The advisor built the Canvas app for the project, although it wasn't his area of expertise. While he found the experience valuable, he expressed uncertainty about whether he had fully optimized the app's development. Additionally, the decision to reuse approximately 80% of the code from the previous app presented challenges. Some portions of the code were overly complex and of suboptimal quality. When new and complex requirements emerged, it was deemed too late to rewrite the

entire code, and he had to work with this less-than-ideal codebase, making it challenging to make alterations due to its initial construction.

In terms of IT tools and information flow, the same internal practices described in Case 1 for planning allocations, logging hours, creating focus actions, and updating status was also done in this case. The advisor acknowledges the need for improvement in these processes but fears that enforcing a streamlined workflow might face challenges with user adoption, particularly for minor or ad hoc tasks that don't require detailed planning or focus actions. Another challenge faced by the advisor is managing communication across various platforms. As an external consultant for the client, he also has an account with them with its own email and Teams account where he would send and/or receive information. On top of these, he also has his Forward Forever email and Teams account where information is also exchanged. This multiplicity of platforms makes it difficult to track where specific information has been shared and information searching becomes time-consuming.

5 Future value stream map

5.1 Proposed solutions

Considering the wastes identified in the current value stream maps, the proposed solutions are (1) optimisation of project management by integrating data entry processes into a unified solution, (2) improving project status visibility and establishing clearer communication practices with customers, (3) automating status updates to promote timely actions when there are diversions to the plan, (4) reducing the frequency of customer meetings yet aligning with client requirements and priorities, and (5) exercising caution when initiating customer projects before the holiday season. The first four proposed solutions are visualised and described in **Error! Reference source not found.**. The final proposed solution necessitates no alteration to the existing systems. Recognizing that holidays are an integral aspect of a healthy work-life balance and employee well-being, the sole adjustment required is the exercise of caution when commencing customer projects before the holiday season. This approach is aimed at minimising disruptions and the need for relearning.

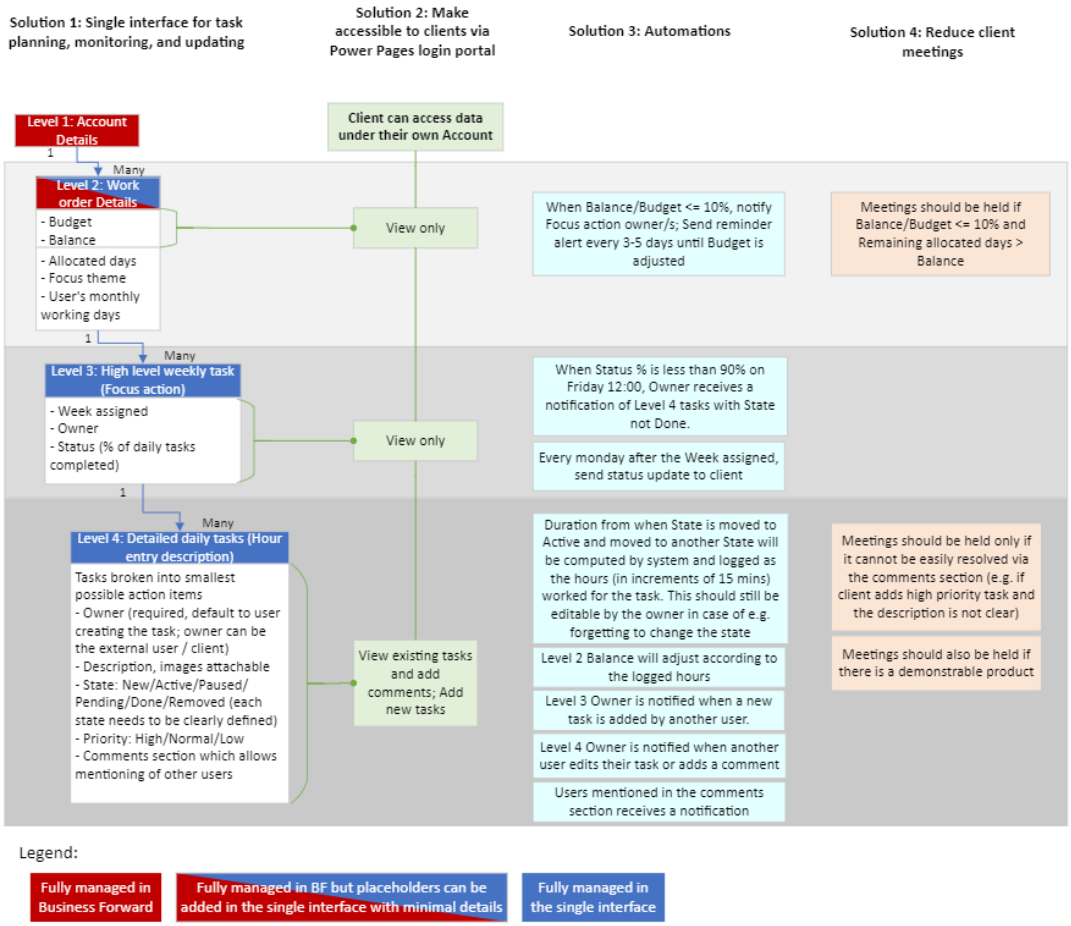


Figure 8. Proposed solutions.

The first proposed solution is the integration of the Hour Entry App and Weekly Meeting App. This integration streamlines task planning, monitoring, and updates by providing a single interface. This interface follows a structured hierarchy akin to the Scrum methodology: "themes" represent customer accounts, "epics" signify work orders, "user stories" correspond to focus actions, and "tasks" equate to hour entries. Within this unified interface, internal users gain the ability to view existing work orders and create new ones for allocation purposes. However, it's essential to note that the addition of new work orders that have undergone approval will still be managed within Business Forward, as is the current practice. All the fields present in the Work Order table, currently in use, will continue to be relevant and available, particularly in Business Forward. This includes work order budgets which will be managed in Business Forward but viewable in the single interface. Balances (budget minus actual hours worked) and number of working days monthly should also be visible for easier planning and allocation of workdays for upcoming months in that particular work order. Additionally, a new choice field,

“Focus theme,” will be integrated into the work order table, which was previously stored separately. The focus themes represent the values of the company and all focus actions need to be aligned with at least one of these focus themes. The researcher finds it best to associate this at the work order level instead of the focus action level since the current trend is that all customer work fall under the same focus theme, and most internal work are then associated with the other focus themes.

The focus action will then serve as a high-level weekly task under the work order with a week date range assigned to it. Each focus action is owned by a single user who will be responsible for its completion. Monitoring it will be made easier through a Status bar which is a rollup calculation of the daily tasks State (i.e. if all daily tasks’ State under the focus action are marked Done, then the focus action Status is 100%). The suggested State choices for the daily tasks are New (tasks that haven’t started), Active (tasks that are currently being done), Paused (tasks that are on-hold due to unavailability of the user such as when the task remains undone but the user goes for lunch break), Pending (tasks that are on-hold because of other users being unavailable), Done (task has been completed), and Removed (task has been deemed unnecessary or task has been jettisoned to accommodate other tasks).

Daily tasks are the smallest possible action items relevant to the focus action. Each daily task will also have their own owners in case there are multiple people collaborating in the same focus action. Owners will, by default, be the user who created the daily task, but can be changed if necessary. There can only be one owner per daily task as the hours spent on that task will serve as the work hours of that individual. The user can add a free-form text description related to the task and also attach media files to it in order to make the task as clear as possible. Meeting notes can also be stored in the description section in order to reduce going back and forth with OneNote. The task can also be labelled as either High, Normal, or Low priority which will be important in renegotiating budgets should it arise. The default value for task priority is Normal unless the user changes it. Lastly, all users are allowed to add comments to daily tasks, including those belonging to other users. They can also mention other users in the comments.

The comments section should help lessen the need for meetings and also have a quick reference for what has been agreed or disagreed.

The second proposed solution is allowing customers to view the status of the project. A potential tool that can be used for this is Microsoft Power Pages wherein specific individuals (ideally the Business Owner) in the client organisation can log in, view the status of the work order, focus actions, and daily tasks under the Account where they belong to, and add comments to it. In addition, they can be given rights to add new daily tasks if there is a specific feature that they need or if they have spotted a bug in the app. Ideally, they should only be able to add tasks to future weeks' focus actions and not the current one. They should also choose the level of priority for this task so the developer can know if it's business-critical or just a good-to-have feature. These roles and responsibilities should be discussed with the client at the onset, ideally during the project kick-off meeting. During this meeting, it should also be decided which platform will be used for all project-related communication. Ideally, the client should be encouraged to use the comments section of this Power Pages platform but if they find another platform to better fit the purpose, then everyone should know to use that instead to avoid unnecessary duplication of discussions across various platforms.

The third proposed solution involves automations. On the work order level, it would be ideal to have a notification when the balance/budget ratio is equal to or less than 10%. The notification should be sent to the focus action owner/s who can then assess if the project is nearing completion, or if more budget needs to be requested, or if certain features/tasks need to be jettisoned. Ideally, another notification should trigger after a specified period (3-5 days) if the balance/budget remains equal to or less than 10%. On the focus action level, a notification sent to the focus action owner should trigger every Friday at 12:00 if the status is less than 90%. The notification can include a list of daily tasks whose state is not Done. This allows the user to rectify potential mistakes (e.g. forgetting to mark the task as Done) or move the task to the next week if there is not enough time to complete it during the current week. In the following week's Monday at around 9:30, a status update should automatically send to the client which lists all the relevant daily tasks scheduled in the previous week and their state as of Monday

morning, around 9:00. This will serve as their weekly status update regarding the ongoing projects. Finally, on the daily task level, there are five automations that can be made. To lessen the user's work load, the system should be able to compute the duration of time (in increments of 15 minutes) that their task was in the "Active" state and create hour entries based on this, subject to the user's confirmation. The user can still make edits in case they have forgotten to change the task's state and need to make adjustments to the duration. Once the user confirms the hours, the work order balance should also adjust accordingly. To ensure that the focus action owner is aware of the tasks added by other users, a notification should be sent to the focus action owner regarding the new daily task added by another user. When there are changes made or comments added to a task by a user other than its owner, then a notification should also trigger to alert the daily task owner. Lastly, the comments should allow tagging or mentioning of other users (in case their input is required for the task) and a notification should then be sent to that mentioned user accordingly.

The implementation of the initial three proposed solutions paves the way for the realization of the fourth proposed solution which is to lessen client meetings yet maintain communication and ensure updates are given on a timely manner. Nevertheless, it will still be inevitable to schedule meetings with the customers but should only be done when there is a real need for it. Some instances where meetings are needed are when there is a need to renegotiate the budget, high priority tasks are added but specs are not clear and not easily resolved via comments, and when there is a demonstrable product.

Worth noting, these proposed solutions for addressing the identified wastes come with no assured effectiveness. Achieving improvement is an iterative process, and the success of these solutions depends on real-world feedback and adaptability. Remaining open to feedback, both before and after implementation, is crucial for fine-tuning the approach and ensuring that the efforts lead to meaningful enhancements. Ongoing assessment and a commitment to adaptability can drive greater efficiency and excellence in software development practices.

5.2 Improvements in the map

Assuming successful implementation of the proposed solutions and taking recent company events into account, the future state value stream maps are shown in **Error! Reference source not found.** for case 1 and **Error! Reference source not found.** for case 2 with kaizen bursts highlighting the driving factors behind the changes.

5.2.1 Case 1 future value stream map

Error! Reference source not found. reveals significant enhancements in workflow, information flow, and timeline. Key distinctions between the current and future value stream maps include (1) eliminationn of customer-related arrows signifying status meetings and clarifications, replaced by interactions through the single interface, (2) replacement of the Hour Entry App with the single IT interface, (3) exclusion of OneNote from the map, (4) consolidation of certain process blocks into a single entity, (5) improved lead time and process time in most process blocks, and (6) enhanced activity ratio, indicating increased efficiency. The kaizen burst associated with the initial process block suggests a post-holiday initiation meeting, merging it with account test planning. Despite the absence of customer-provided access rights, automated reminders expedite the process, reducing lead time to an estimated 7 days.

In the POC Development phase, improved project status visibility through the single interface (accessible to external users through Power Pages) leads to reduced status meetings and decreased task switching. Process time estimates are lowered accordingly, with a lead time of 20 days, assuming 4 hours of daily work by advisors. Accounting for holidays and weekends, this result is multiplied by 1.5 and rounded up to the nearest tenth. Similar calculations are applied to other process blocks, except those requiring external collaboration. Work post-POC architecture review resumes post-holidays to mitigate relearning and waiting for relevant individuals to return. The holiday period, approximately 130 days, impacts the timeline but not the lead time for POC architecture review.

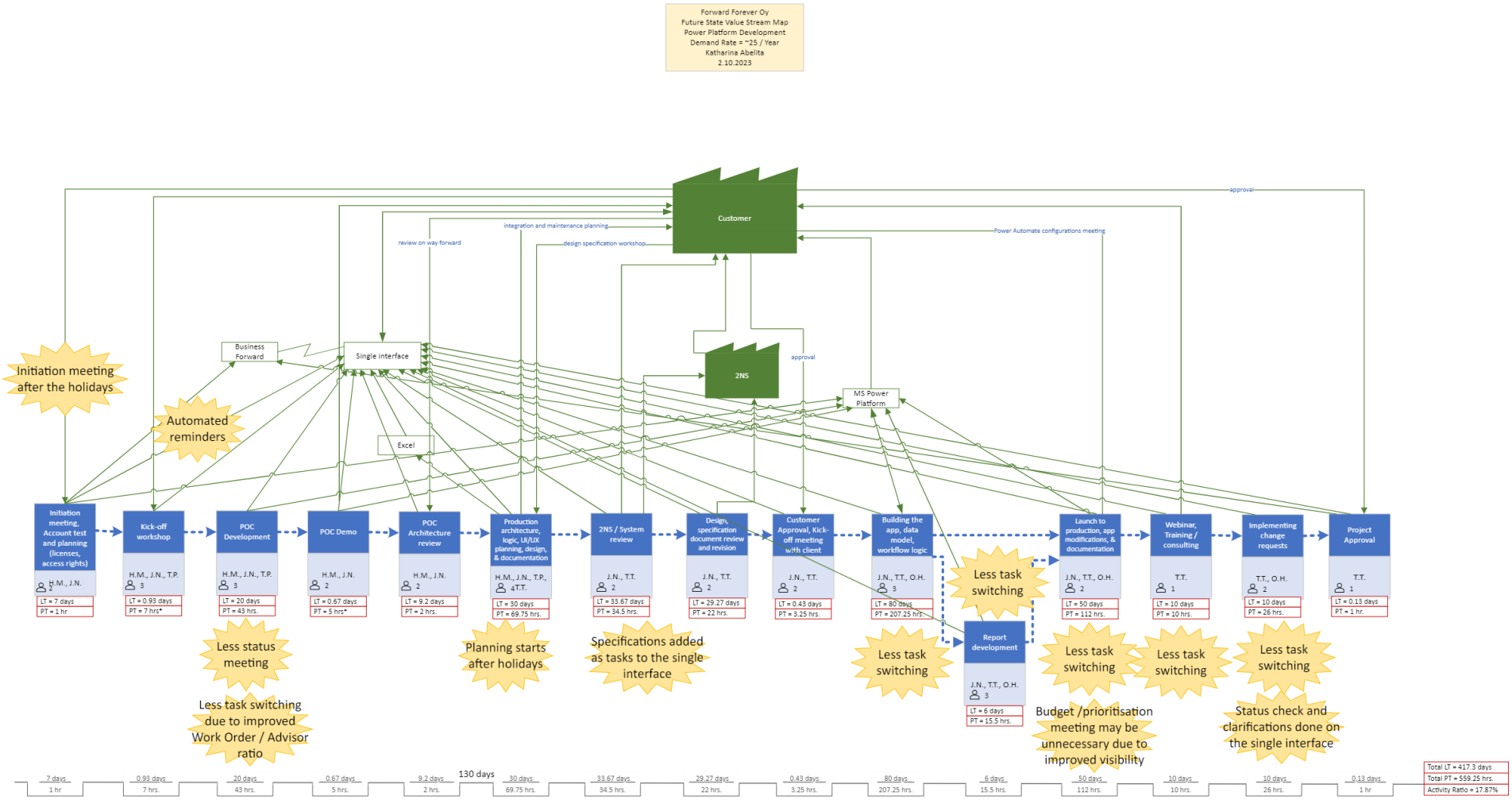


Figure 9. Case 1 future value stream map.

Meanwhile, the open specification meeting process block that occurred after the planning phase is eliminated, as specifications can be added to the single interface, with meetings held only if necessary. During the app-building phase, lead time and estimation calculations remain consistent across subsequent tasks. Report development is incorporated into the timeline due to reduced task switching and shared personnel with app development. Budget and prioritization meetings, as well as change request implementation, may be bypassed with continuous project status updates, aligning with new process times and lead times. These adjustments are expected to improve the total lead time to 417.3 days, with a total process time of approximately 559.25 hours, resulting in an activity ratio of 17.87%.

5.2.2 Case 2 future value stream map

In **Error! Reference source not found.**, the Future Value Stream map for the second case showcases notable enhancements: (1) the introduction of a kick-off workshop preceding development work, (2) integration of the mid-development use-case definition process block into the kick-off meeting, (3) the replacement of the Customer side Teams channel and internally used Hour Entry App and Weekly Meeting App with a unified single interface, and (4) marked improvements in total lead time and activity ratio.

In this envisioned future state, the kick-off workshop will primarily focus on identifying project requirements, specifying details, and recognising essential needs. The workshop serves as a platform for defining roles, responsibilities, and stakeholders, as well as establishing scope parameters. Emphasizing the adoption of a singular communication platform and precisely defining project use cases is pivotal for promoting efficient collaboration and resolving communication challenges. At the workshop's conclusion, meticulous planning of action items aims to unite team members and stakeholders in a collective understanding of the project's objectives. These action items are recorded within the single interface, and automated reminders are set to prompt responsible individuals about pending tasks at designated intervals.

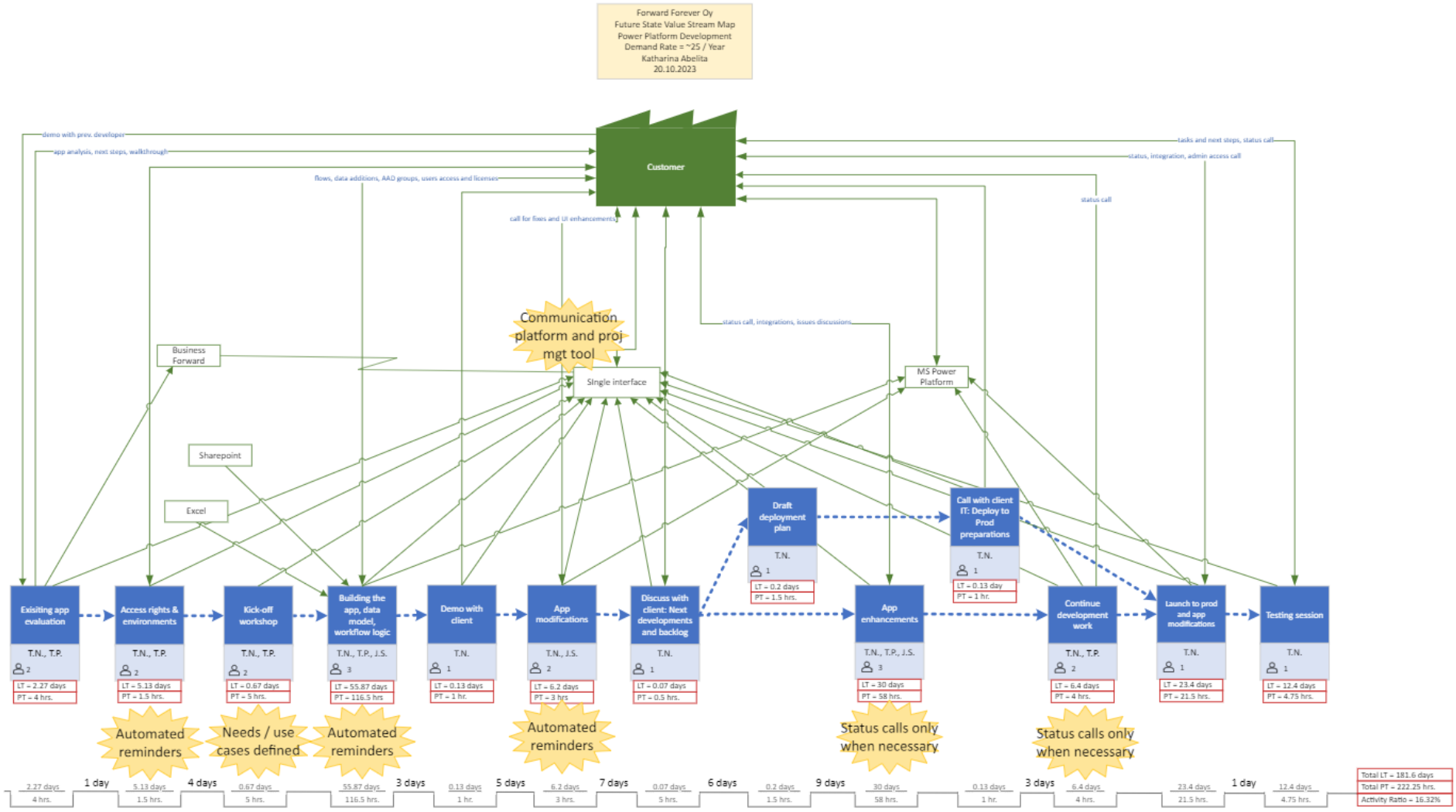


Figure 10. Case 2 future value stream map.

Similar to the first case, the single interface continues to function as a versatile project management tool, allowing advisors to plan their work and monitor progress. Clients gain easy access to project-related information, enabling them to contribute new insights and requests. Consequently, the need for status calls is reduced, as real-time project updates are readily available to clients.

With the implementation of proactive reminders to address specific requirements before the onset of the summer season, waiting times are significantly decreased. As a result, the total lead time is anticipated to decrease substantially, moving from 354.93 days to 181.6 days. Additionally, this process refinement is expected to lead to a remarkable enhancement in the activity ratio, surging from 8.16% to 16.32%.

6 Limitations

This study primarily focuses on the low-code software development process, specifically within the context of Power Apps development. As such, the findings and proposed solutions may not be universally applicable to all software development methodologies or platforms. Additionally, the research's focus on specific stages of development, such as access rights granting and work planning, may limit the generalisability of the proposed solutions to broader development scenarios.

In terms of data collection, this study relied significantly on hour entries, thematic analysis, and interviews with lead developers. These data sources, while providing valuable insights, might lack a comprehensive representation of the entire software development process. With the heavy reliance on the experiences and perceptions of lead developers, the insights gained might be influenced by individual perspectives, biases, or organisational practices. Meanwhile, limitations in access to certain data, particularly those involving clients or other external stakeholders, could restrict a complete understanding of the process and potential bottlenecks.

The proposed solutions offered in response to identified challenges are based on the specific findings and context of the study. Implementing these solutions in

different organisational settings or varied software development environments may yield differing results. The level of applicability and effectiveness of these solutions across diverse scenarios remains untested and would require further validation and adaptation to specific contexts. In addition, the financial implications of the proposed solutions were not covered by this research and thus, also require further consideration.

7 Conclusion

In conclusion, this thesis has presented a comprehensive exploration of the low-code software development process, with a particular emphasis on revealing the inefficiencies inherent in the pre-development phase. Through a meticulous analysis of two illustrative case studies, challenges were found related to securing access rights, and other external parties-dependent information. The main issue that was stemming internally was the current practice of project planning, tracking, and communication which depended on multiple interfaces.

Notably, the use of the Value Stream Map (VSM) has emerged as a valuable tool in identifying and addressing these process inefficiencies. Through VSM, organisations can gain valuable insights into lead times, activity ratios, and points of waste accumulation, fostering data-driven decision-making. The solutions proposed hold the potential to bring about substantial improvements in the low-code software development process. However, the true effectiveness of these strategies can only be gauged through real-world implementation and the continuous collection of feedback. Detailed financial implications, such as Power Pages licensing, require further exploration to ensure the feasibility and efficacy of the proposed solutions.

As the low-code development landscape continues to evolve, this research contributes to the discourse on process enhancement in this ever-changing field. The pursuit of more efficient software development is a perpetual journey, and this thesis adds to the ongoing dialogue in this pivotal area, leveraging the power of the Value Stream Map to drive tangible improvements and waste reduction in the low-code software development process.

8 References

AWS. (n.d.). What Is SDLC (Software Development Lifecycle)? Retrieved from https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20(SDLC,expectations%20during%20production%20and%20beyond 16.4.2023

Abdulmalek, F. A., & Rajgopal, J. (2007). Analyzing the benefits of lean manufacturing and value stream mapping via simulation: A process sector case study. *International Journal of Production Economics, 107*(1), 223–236. https://doi.org/10.1016/j.ijpe.2006.09.009

Alahyari, H., Gorschek, T., & Berntsson Svensson, R. (2019). An exploratory study of waste in software development organizations using agile or lean approaches: A multiple case study at 14 organizations. *Information & Software Technology, 105*, 78–94. https://doi.org/10.1016/j.infsof.2018.08.006

Anand, G., Chandrashekar, A., & Narayanamurthy, G. (2014). Business Process Reengineering Through Lean Thinking: A Case Study. *Journal of Enterprise Transformation*, 4. doi: 10.1080/19488289.2013.879681.

Coleman, G. & O'Connor, R.V. (2008). An investigation into software development process formation in software start-ups. *Journal of Enterprise Information Management, 21*(6), 633–648. https://doi.org/10.1108/17410390810911221

Creswell, J. W. (2014). Research design: qualitative, quantitative, and mixed methods approaches. Sage publications.

Denzin, N. K., & Lincoln, Y. S. (2011). The SAGE handbook of qualitative research. Sage publications.

Dooley, J. (2011). Software Development and Professional Practice. Springer.

Gunaki, Pradip & Devaraj, S. & Patil, Shanawaz. (2021). Process optimization by value Stream Mapping. Materials Today: Proceedings. 54. 10.1016/j.matpr.2021.08.304.

Hines, P., & Rich, N. (1997). The seven value stream mapping tools. *International Journal of Operations & Production Management, 17*(1), 46–64. https://doi.org/10.1108/01443579710157989

Karvonen, T., Behutiye, W., Oivo, M., & Kuvaja, P. (2017). Systematic literature review on the impacts of agile release engineering practices. *Information & Software Technology, 86*, 87–100. https://doi.org/10.1016/j.infsof.2017.01.009

Khurum, M., Gorschek, T., & Wilson, M. (2013). The software value map — an exhaustive collection of value aspects for the development of software intensive products. *Journal of Software: Evolution and Process*, 25, 711-741. https://doi.org/10.1002/smr.1560

Khurum, M., Petersen, K., & Gorschek, T. (2014). Extending value stream mapping through waste definition beyond customer perspective. *Journal of Software: Evolution and Process, 26*(12), 1074–1105. https://doi.org/10.1002/smr.1647

Martin, K., & Osterling, M. (2014). *Value stream mapping: How to visualize work and align leadership for organizational transformation*. McGraw-Hill.

Middleton, P., & Joyce, D. (2012). Lean Software Management: BBC Worldwide Case Study. *IEEE Transactions on Engineering Management, 59*(1), 20-32. https://doi.org/10.1109/TEM.2010.2081675

Mäntyomena, H. (2020). How to modernize internal processes with Power Platform. *Forward Forever Blog*. https://forwardforever.com/how-to-modernize-internal-processes-with-power-platform/

Nowell, L. S., Norris, J. M., White, D. E., & Moules, N. J. (2017). Thematic Analysis: Striving to Meet the Trustworthiness Criteria. *International Journal of Qualitative Methods*, *16*(1). https://doi.org/10.1177/1609406917733847

Poppendieck, M., & Poppendieck, T. D. (2007). *Lean software development: An agile toolkit* (12th printing 2007.). Addison Wesley.

Shingō, S. (1989). A study of the Toyota production system from an industrial engineering viewpoint (Rev. ed.). Productivity Press.

Singh, B., Garg, S.K., Sharma, S.K. and Grewal, C. (2010), Lean implementation and its benefits to production industry. *International Journal of Lean Six Sigma*, 1(2), 157-168. https://doi.org/10.1108/20401461011049520

Staats, B.R., Brunner, D.J. & Upton, D.M. (2011), “Lean principles, learning, and knowledge work: evidence from a software services provider”, *Journal of Operations Management, 29*(5), 376-390.

Wang, S., Tang, J., Zou, Y., & Zhou, Q. (2020). Research on production process optimization of precast concrete component factory based on value stream mapping. *Engineering Construction & Architectural Management (09699988), 27*(4), 850–871. https://doi.org/10.1108/ECAM-10-2018-0455

Structure of the verification and interview meeting

a. Walkthrough of the data collection and analysis methods
b. Clarification of certain information that could not be inferred from the data collected (e.g. What was the webinar and consulting about, what was the project management tool used)
c. Presentation of the current value stream map
d. Asking feedback if the map is accurate or needs revising and on which areas
e. Discussion of the experiences of the interviewee regarding the project
f. Discussion of root causes of the inefficiencies identified
g. Asking about lessons learned and process improvement ideas

Consent form for research participants

Research Participant Consent Form	
Title of study:	Analyzing the Efficiency of Software Development Process in a Low-Code Technology Company: A Case Study Using Value Stream Mapping
Researcher:	Katharina Abelita

You are being invited to participate in a research study that aims to identify wastes in the low-code software development process using a Value Stream Map and create a plan for process improvement and waste elimination.

Study Procedures

If you agree to participate in this study, you will be asked to:

- Participate in a semi-structured interview with the researcher, which will be recorded.
- Allow the researcher to investigate the software development process, particularly of the case/s selected, at your organization or workplace.

The initial semi-structured interview will take approximately 30 minutes and a follow-up interview of approximately 30 minutes may be conducted to verify the interpretations made.

Benefits and Risks

There are no direct benefits to you for participating in this study. However, your participation will contribute to the understanding of the current state of the software development process in your organization and in the formulation of improvement proposals. The risks of participating in this study are minimal and include possible discomfort or inconvenience during the interview process.

Confidentiality

This study will be published online via theseus.fi. The study findings and some information collected during this study may then be readily accessible by the public. Any personal information obtained will be de-identified to protect your and your client's privacy.

Voluntary Participation

Your participation in this study is voluntary, and you may withdraw from the study at any time without any consequences.

Consent

By signing this form, you acknowledge that you have read and understood the information provided in this consent form, and you agree to participate in this study.

If you have any questions or concerns about this study, please feel free to contact the researcher at katharina.abelita@edu.karelia.fi.

Participant Name and Signature:	This part redacted	Date:	15.09.2023 09:27:51 (UTC +0300)
Researcher Name and Signature:	Katharina Abelita Katharina Abelita	Date:	15.09.2023 09:11:32 (UTC +0300)

Research Participant Consent Form			
Title of study:	Analyzing the Efficiency of Software Development Process in a Low-Code Technology Company: A Case Study Using Value Stream Mapping		
Researcher:	Katharina Abelita		

You are being invited to participate in a research study that aims to identify wastes in the low-code software development process using a Value Stream Map and create a plan for process improvement and waste elimination.

Study Procedures

If you agree to participate in this study, you will be asked to:

- Participate in a semi-structured interview with the researcher, which will be recorded.
- Allow the researcher to investigate the software development process, particularly of the case/s selected, at your organization or workplace.

The initial semi-structured interview will take approximately 30 minutes and a follow-up interview of approximately 30 minutes may be conducted to verify the interpretations made.

Benefits and Risks

There are no direct benefits to you for participating in this study. However, your participation will contribute to the understanding of the current state of the software development process in your organization and in the formulation of improvement proposals. The risks of participating in this study are minimal and include possible discomfort or inconvenience during the interview process.

Confidentiality

This study will be published online via theseus.fi. The study findings and some information collected during this study may then be readily accessible by the public. Any personal information obtained will be de-identified to protect your and your client's privacy.

Voluntary Participation

Your participation in this study is voluntary, and you may withdraw from the study at any time without any consequences.

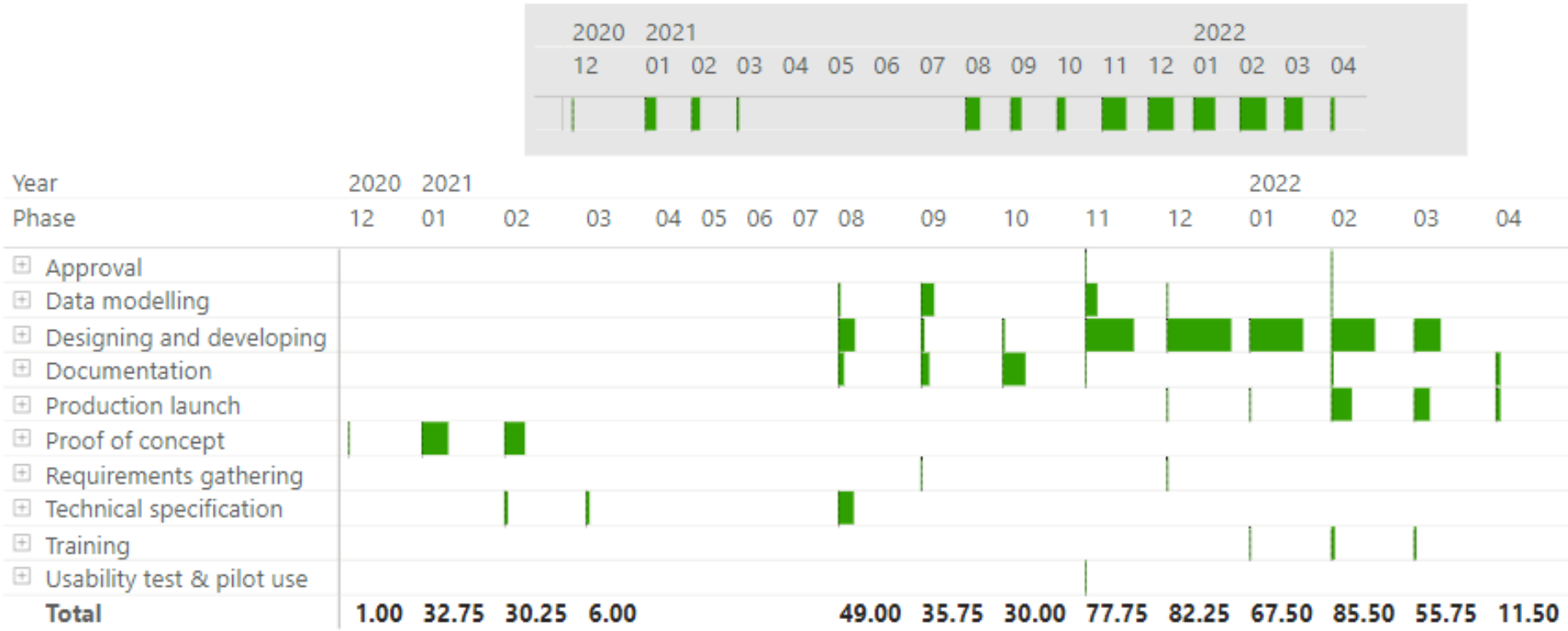
Consent

By signing this form, you acknowledge that you have read and understood the information provided in this consent form, and you agree to participate in this study.

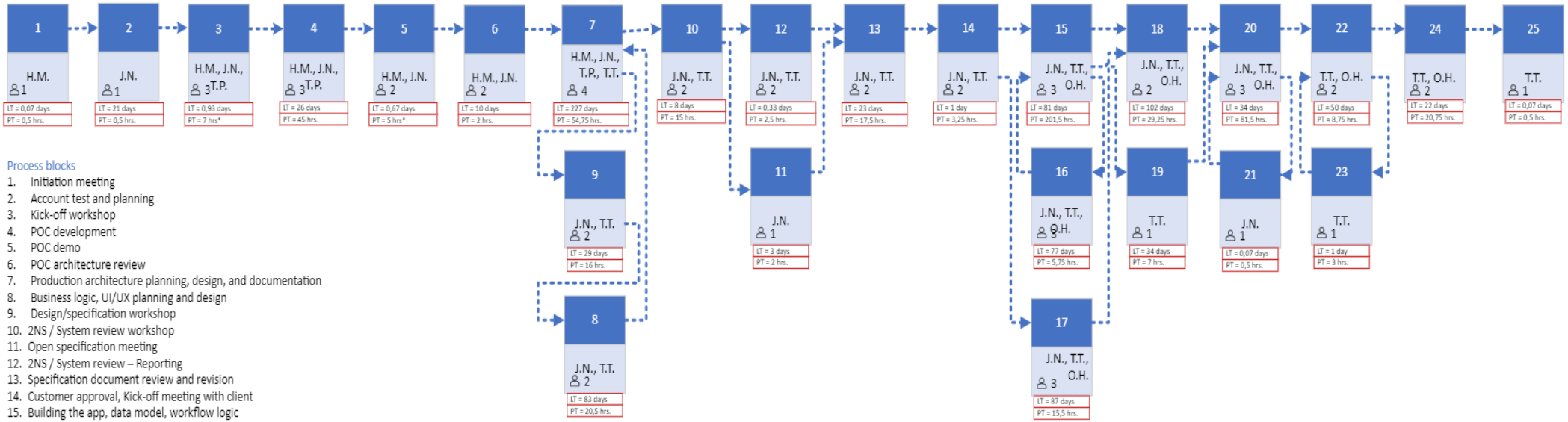
If you have any questions or concerns about this study, please feel free to contact the researcher at katharina.abelita@edu.karelia.fi.

Participant Name and Signature:	This part redacted	Date:	12.10.2023 10:16:01 (UTC +0300)
Researcher Name and Signature:	Katharina Abelita Katharina Abelita	Date:	12.10.2023 09:40:30 (UTC +0300)

Case 1 workflow monthly process time visualisation using Power BI



Case 1 workflow draft using initial set of process blocks



Process blocks

1. Initiation meeting
2. Account test and planning
3. Kick-off workshop
4. POC development
5. POC demo
6. POC architecture review
7. Production architecture planning, design, and documentation
8. Business logic, UI/UX planning and design
9. Design/specification workshop
10. 2NS / System review workshop
11. Open specification meeting
12. 2NS / System review – Reporting
13. Specification document review and revision
14. Customer approval, Kick-off meeting with client
15. Building the app, data model, workflow logic
16. Testing and debugging
17. Report development
18. Launch to production
19. Webinar
20. App modifications
21. Budget and prioritization meeting with client
22. Documentation for handover
23. Training / consulting
24. Implementing change requests
25. Project Approval