



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



UNIVERSITY OF GOTHENBURG

---

# Managing Software Development in a Small Non-IT Company – A Case Study

Bachelor of Science Thesis in Software Engineering and Management

MARKUS EMILIO PUERTO GUTIÉRREZ  
MARKUS JÄRVELÄINEN



The Author grants to University of Gothenburg and Chalmers University of Technology the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let University of Gothenburg and Chalmers University of Technology store the Work electronically and make it accessible on the Internet.

## **Managing Software Development in a Small Non-IT Company – A Case Study**

© MARKUS EMILIO PUERTO GUTIÉRREZ, June 2023.

© MARKUS JÄRVELÄINEN, June 2023.

Supervisor: PALLE DAHLSTEDT

Examiner: BIRGIT PENZENSTADLER

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

# Managing Software Development in a Small Non-IT Company – A Case Study

Markus Emilio Puerto Gutiérrez  
Department of Computer Science and  
Engineering  
University of Gothenburg  
Gothenburg, Sweden  
guspuema@student.gu.se

Markus Järveläinen  
Department of Computer Science and  
Engineering  
University of Gothenburg  
Gothenburg, Sweden  
gusjarmap@student.gu.se

**Abstract**—We have done a case study on a small sheet metal construction company in Sweden that is developing their own software that will be beneficial for their business. We interviewed staff from different roles in the company and performed thematic analysis on the data we collected. This study might show what is needed from the management of a small non-IT company to be able to manage both their main business and the additional activity of software development. It might also show beneficial habits in the software development process in a company like this. We could not come across many examples of very similar cases, which made us believe that this topic needed to be studied. We found that in this company there are many things that contribute to making it possible for the dual activity to take place. For example, the varied skills that some of the staff members had made it easier for them to take on different tasks, and also that the managers had the will to see the software succeed despite several drawbacks. We hope that our findings might be useful for other companies that want to develop software, or for future research work that includes more companies.

**Keywords**—*software development, small non-IT company, company management, work environment*

## I. INTRODUCTION

Working with software development in a small non-IT company could be assumed to be complicated because they likely need to put high focus on their primary business. Software development introduces a second activity that also takes time and resources, and for a small company, deviating from the main business can presumably be risky and require a well-managed approach. However, software solutions (for example mobile apps, websites and other programs) have become more and more popular since companies might need solutions that are tailored for them and that may not exist in the market, and there may also be a need to continuously maintain and further develop the solutions over time. Hence, the purpose of our study is to find both software development habits and managerial habits, in a small non-IT company that

we hope will be useful for other companies in similar situations.

In our research, we consider a non-IT company one where their main business isn't in IT-related areas (for example software development). It does not, however, mean that the company isn't using software. We found a small sheet metal construction company outside Gothenburg, Sweden that has managed to develop software solutions for themselves, and therefore we determined that a case study on this particular company was a good fit to our purpose. Hereinafter we will refer to it as being a construction company that we call Company A because one of the co-owners has asked us not to name the company, and therefore we decided also not to disclose its precise location. We believe this company can give this study some useful information as they decided to take on the dual activity of working with software development alongside their primary business, and they seemingly succeeded in doing this. Construction companies would fall under the category of being a non-IT company, as they presumably do not usually develop software, especially small ones. This is especially fitting as it has been found that the construction industry has not had major transformations in terms of embracing new technologies [1].

We therefore think that a study like this could be useful as the results could be taken into consideration by other small non-IT companies that have a need for software solutions.

## II. RELATED WORK

There has been related work done on non-IT companies, and we found one study where the smallest company is defined as having less than 5000 employees [2]. We do not know the exact size of that company, but considering how the employee ranges are defined we assume that it is not close to the small size of the company we have chosen to work with. In the same study [2], one finding was that balancing IT development between making sure the systems in place are as

efficient as they should, and using IT in an innovative way can be very hard for non-IT companies.

Also, regarding small (and medium-sized) enterprises, it's more difficult for them to develop software than larger business due to the lower number of resources they have available [3]. Since it has been argued that smaller companies have a tougher time developing software than larger companies, we believe this is relevant to our study since Company A is a small company that is developing software. Furthermore, it is also mentioned that unlike bigger businesses, smaller companies have the ability to focus on the actual value that they will get from the innovation [3]. This is related to our study because it shows that despite Company A being a small company, they could have an advantage that larger companies don't have.

The term non-IT company is broad, but it has been found that specifically the construction industry seems to have been slower to implement new technology, which includes digitalization, and it would surely profit if they started to use these technologies [1]. At the same time, it is not that clear how this change would take place since different companies have different issues that may need to be solved in ways that are specific to them, and therefore the responsibility for the transformation is on each individual company [1].

It is considered that there is very little information on how digitalization is approached by small companies [4], which motivates our study. It seems that results from related work regarding small companies found that the force for change, in their digitalization processes, is started and maintained by the leaders of the company, and not by the employees themselves [4], this is why we think it is important to also study managerial habits.

### III. RESEARCH METHODOLOGY

Considering that small non-IT companies might not be willing or even capable of taking the risk of spending time and resources on a secondary activity, such as software development, and since there might be a need for it, we have chosen the research questions below to extract useful information for the case study that could be used to help other companies of this kind.

**RQ1:** How could a small company outside the IT industry manage their software development while still prioritizing their main business?

**RQ1.1:** What managerial or financial risks could be involved to be able to develop software in such companies?

**RQ2:** What habits are beneficial in small companies outside the IT industry for them to develop their own software successfully?

**RQ2.1:** How is the work environment affected by the dual activities involved when a company manages software development while also managing their main business?

**Methodology for this study** – Here we present what methodology we will use, the reason for it, and we also discuss an alternative.

To answer these research questions, we did a case study on Company A, on how they handle their software development, to find information about what did and did not work well for them. Then we did literature research to understand common approaches to software development in small non-IT companies, and finally we compared these results with the information we got from the case study.

As for alternative methodologies, ideally, conducting a survey of managers and developers across many different small non-IT companies would give a broad representation of companies in that category. However, unlike the small construction company we are doing a case study on, Company A, it seems rather uncommon for small non-IT companies to take on software development as a secondary, simultaneous task. This could lead to scarce availability of similar companies to survey. Furthermore, identifying such companies would be time consuming and not feasible for a bachelor thesis on this topic. Therefore, we decided to go with qualitative research in the form of a case study with interviews. This way we would get answers with more information considering that we only studied one company.

Another reason we decided to do a case study is that we studied a company that is developing their own software instead of using software entirely developed by another company, and a case study is deemed an appropriate methodology given this fact [5]. Furthermore, our selection of the case was not done by sampling but rather a purposeful choice since we considered this case to be unusual, and a case study would be a good choice for a unique situation [5].

**Interviews** – The case study was carried out using interviews with managers, developers and non-IT employees in Company A. Non-IT employees refer to employees in this company that have construction-related tasks rather than software development, not necessarily that they do not use IT tools. The case study is of the exploratory type since we have no hypothesis. Instead, we reached our hypothesis after our investigation. We planned on conducting semi-structured interviews and we planned to have questions that are both general for all interviewees and questions designed for the interviewee roles in Company A. The reason why we chose interviews was because we want open discussion with the employees to collect our data. Interviews would be a good way to achieve this [5]. We decided that we were going to be two interviewers conducting the interviews as it has been shown that two interviewers are likely to ask more questions [6]. We believed that if more questions were asked, then we would get more valuable information.

**Thematic analysis** – After having collected all of our data from the case study in Company A, we analyzed the data to get the answers about how the small company outside the IT-business managed the software development, and also what qualities were useful in their development process. We did this by following the iterative steps for thematic analysis as described in an article that we found [7]. As presented in Table 1 below, these are the steps from that article:

**Table 1**  
**The six steps of thematic analysis**

Step	Name
1	Familiarizing yourself with the data
2	Generating initial codes
3	Searching for themes
4	Reviewing themes
5	Defining and naming themes
6	Producing the report/manuscript

**Literature research** – We first searched for papers that were discussing software development in either small companies, non-IT companies or both combined. We used those to come up with initial theories regarding our topic. The next step was to collect information about research question 1 and its sub-question, so we searched for papers that focus on the managerial aspects in software development in small companies and companies outside of the IT-business. As for research question 2 and its sub-question, we searched for papers that are about software development in small companies and in non-IT companies and try to identify habits. We included these findings when we analyzed and discussed the results from the case study.

#### IV. METHODS USED

In this section we describe how we put the methodology to use in our research.

**Interview data collection** – First, we describe how we carried out the interviews to collect our data:

We planned to, as a first option, to record and transcribe the interviews. We asked for permission before the interview started, and explained to the interviewee that the recording would only be used for working with the answers. It turns out we were allowed to record all the interviews. Since we interviewed staff at a Swedish company, we offered the interviewees to choose between being interviewed in Swedish or English, in case some of them were not used to speaking English or if they felt more comfortable with Swedish. For the interviews that were held in Swedish, we needed to put time into translating the interview into English since only one of the researchers is fluent, and the other only has basic skills in Swedish. As for transcribing, it was shown in a study that transcribing one-hour interviews would take about eight hours on average [6], with this information we estimated to spend one week to transcribe all the interview recordings we collected. This is based on the fact that we planned on interviewing the following staff:

- Two of the co-owners of Company A, who also manage or make decisions in the company.
- A developer that has been doing work in this company, as well as one of the managers that has experience in software development and took part in the development in the company, we interviewed that person twice, in both the manager role and developer role.
- At least one non-IT employee in the company.

As mentioned in the Introduction section, a co-owner of Company A asked us to not mention the company’s real name, we therefore decided that it made sense to not use the real names of the interviewees. We will therefore refer to them as presented in Table 2 in the Results section.

We planned on having at least a total of five interviews, with four interviewees. We expected each interview to last more than 30 minutes, but less than an hour.

We found that it was recommended to avoid “yes”, “no” or questions starting with “why”, in semi-structured interviews [6]. Given this information we designed our questions with this in mind, reducing such questions to a low amount, and designing all questions so that they require answers that provide more reasoning. After we had a draft of the interview questions, we went over them again, keeping in mind that according to this paper [6], questions that are too detailed might make them harder for the interviewees to answer. We therefore changed some sub-questions to make them less detailed, and also eliminated some sub-questions altogether. It was important for us to get valuable answers to the questions we had, therefore we used probes if we do not feel like we are getting enough information, as probes are supposed to help expand the answers [6].

In the same paper it was mentioned that some questions about competence and experience might be perceived as intrusive to the interviewee and therefore could make the interview feel uncomfortable [6]. We therefore needed to remove or rephrase some questions about the interviewee’s background that could be considered sensitive. This resulted in some of our questions being more open, which would allow the interviewees to answer as much as they felt was necessary, instead of directly asking them for a specific answer.

**Interview data processing** – Here we explain how the data we gathered during the interviews was processed:

As for transcribing, since we had the interviews recorded, we could go back and listen to parts of the interviews multiple times to make sure they were transcribed as accurately as they could be. The author who was fluent in Swedish got assigned the interviews of Manager 2 and the Non-IT staff member, whose interviews were held in Swedish and needed to be translated after transcribing. The other author transcribed Developer 1’s, Developer 2’s, and Manager 1’s interviews, since they were held in English. Even though the interviews were recorded with two separate devices placed at different angles from the interviewee, some parts were unintelligible. However, these parts seemed to mostly be filling silence and held no key information. Due to delay caused by one of the authors getting sick (explained further in “Limitations” below), the author who is not fluent in Swedish helped translate, however the fluent author verified the translation afterwards.

Even though all interviews provided us with useful information for our research project, some interviews went a bit out of track by moving into subjects that we realized would not help us in answering our research questions. After careful evaluation, we decided not to code these parts. For example, during the question where one of the managers was asked to describe their experience as a manager, the manager answered

the question, and then expanded by talking about management in general and the manager’s opinion on leadership and that the manager believes there are many management buzzwords out there.

**Coding process** – Next, we will go over how the coding of the transcripts was performed:

The five interviews were coded in collaboration. Since we were the ones performing the interviews, and listened to the recordings repeated times, and subsequently transcribed them, we had already become quite familiar with the data set. However, since each author transcribed different recordings, we were more familiar with some transcriptions than others. Since the transcription of the interviews was both delayed and took longer than we had expected we decided that we needed to speed up parts of the thematic analysis by coding together and discussing the codes at the same time instead of each coding separately and then come together to compare our codes. Also, we decided to make all of the five interviews into one big dataset, meaning that the same list of codes would be applied to all of the interviews.

We took a deductive approach to the dataset since we had some ideas and theories on what the data would show. Also, when coming up with most of the interview questions we had in mind our research questions, which means we had an idea as to what the answers would be. Further, during the interviews themselves we heard in the answers parts that seemed to be very relevant to our research, which means that we were already starting to form theories before we started the coding part.

Since there was no interview transcript that we considered to be more important than any other, we decided to go through the interviews alphabetically, starting with the developers, then the managers and finally the non-IT staff. We went paragraph by paragraph, reading through the sentences looking for phrases, sentences or paragraphs. Then, we started coding them with codes that identified the managerial and financial matters, which are related to our first research question and its sub-question, and codes that identified beneficial staff qualities, development process, and workflow methods and tools to help answer our second research question and its sub-question. We decided that the names of the codes were going to be short, since we found that codes are usually only defined with a brief phrase or a single word that defines an attribute found in the data [8].

We used a mix of deductive and inductive approaches, because despite performing the coding process with pre-existing ideas about the dataset, we were able to also take an inductive approach allowing us to expand our set of codes to better label the dataset. During the coding of the first transcript, we found that one topic that was repeated various times was the concerns expressed by the interviewee. Accordingly, we added a code called “Concerns” which we used to label anything that the interviewee felt was negative. As a result of this, we later also added a code called “Contentment” so that we could label anything that the interviewee felt was positive. We thought that by adding this code, we would be able to see what parts of the dual activity

in Company A were perceived to be good by the people involved or affected by it.

We applied the same codes to the two following transcripts, and while coding Manager 1’s interview we noticed that one common topic in the transcript that we didn’t have a code for was the comparison made between Company A and other companies. Therefore, we added a code called “Comparison to other companies” and went through the transcript again to apply it to anything that was related to it, and then we went back to the first two interviews we coded to add that code.

We made sure that no extract from the transcript was labeled with two codes, meaning that if an extract was labeled half “Managerial issues” and half “Concerns”, we would have to pick the code that best fit that extract. Also, we realized that some of the labeled extracts did not fit the code it was labeled and therefore needed to be changed to another code, or even sometimes it did not fit any of the codes we had and therefore was left unlabeled. Further, we realized that some labeled extracts did not make sense because they did not have enough context, so we either expanded the label to give it more context, or if there wasn’t any context that could be given then we left it unlabeled.

**Theming process** – Using an online whiteboard, we put all the coded extracts in their respective places. We then put the eight codes in a different section of the board, and then we discussed which of them belonged together and which did not belong together and began to put them in groups. After having done this, we began to give names to the groups, which resulted in five themes, that covered all of the eight codes. While coming up with the names for these themes we did not have an intention to make the theme names longer than the code names, although it has been said that theme names should be longer than the codes to make them more understandable [8]. Instead, we focused on whether or not the theme names covered the codes in them, because we did not think our codes were ambiguous to begin with. When reviewing these five themes, we moved some of the coded extracts to other codes, and some extracts were removed altogether because we considered that under the theme’s name, they were no longer applicable to their respective codes. We then gave the themes a final look and made some small adjustments to make sure the naming was appropriate to encompass all their respective codes. Finally, we gave these themes a description, and also described why they were important for our study.

## V. RESULTS

In this section we present the results from our data collection.

**Interviews** – As presented in Table 2 below, we describe the people we interviewed.

**Table 2**  
**Interview participants, their roles and their stated backgrounds**

Interviewee	Description
Developer 1 (D1)	Took part in the software development for Company A. Studied Information Architect, bachelor’s

	level, for 3 years. 3 years of professional experience in the role.
Developer 2 (D2)	(Also is Manager 1) Helped with software development and design in the company. Learned web design in university.
Manager 1 (M1)	Co-owner of Company A. Manages the company overall and took most responsibility for the software development. Took a Master's in industrial engineering, and innovation and management.
Manager 2 (M2)	Co-owner of Company A. Manages parts of the company and takes responsibility for production and staff matters. Took two years of engineering subjects at university and is a trained sheet metal worker. Has 6 years of experience in the role.
Non-IT staff member (N1)	A project manager in the construction side of Company A. Education as a real-estate agent, civil economist, and civil engineer. 5 years in the role.

All interviewee participants in our case study were between 30–49 years old at the time of the interview. Specifically, four of the interviewees were between the ages of 30–39, while one of the interviewees' age was between 40–49.

As previously mentioned, one of the managers also helped with software development in Company A. We refer to that person as M1 and D2. Two separate interviews were held with that person and when we interviewed the person as a manager, they mentioned things they did in the role of the developer, and vice versa. We asked that person in both interviews to try and answer the questions from the point of view of the role of the interview questions. However, there were moments when D2 answered as a manager and M1 answered as a developer. We therefore used the information from these two interviews interchangeably for both roles where it was appropriate.

In total, the interviews took roughly 2 hours and 25 minutes for the actual interview parts that were recorded. This excludes the small conversations we had to loosen up the tension in the beginning, the time we took to present ourselves and our topic, and that we asked for recording permission and explained how the recordings were going to be used before we started. We also took some time after the interview to thank for the participation afterwards which is not included in the count.

**Resulting codes** – As presented in Table 3 below, all our identified codes are listed.

**Table 3**  
**Eight codes and their respective numbers**

#	Code
1	Workflow methods and tools
2	Managerial issues

3	Development process
4	Financial issues
5	Concerns
6	Contentment
7	Beneficial staff qualities
8	Comparison to other companies

As presented in Table 4 below, the occurrence of each code (using the same numbers as in Table 3 above) in each interview is listed (the header row is the interviewees, and column header is the code numbers):

**Table 4**  
**Occurrence of codes for each interviewee**

	D1	D2	M1	M2	N1
#1	5	4	4	2	0
#2	24	11	11	20	0
#3	23	18	7	0	1
#4	0	5	6	14	0
#5	17	16	15	25	2
#6	4	5	8	17	2
#7	8	8	12	12	9
#8	7	2	6	3	4

**Resulting themes** – As presented in Table 5 below, we ended up with five themes after all the processing work was finished after the interviews.

**Table 5**  
**Five final themes**

#	Theme	Codes
1	Software development habits	- Workflow methods and tools - Development process
2	Company management responsibilities	- Financial issues - Managerial issues
3	Beneficial staff qualities	- Beneficial staff qualities
4	Staff mood	- Concerns - Contentment
5	Comparison to other companies	- Comparison to other companies

The final themes can be defined as follows:

1. Software development habits: This theme describes any activity that was aimed to establish or maintain a habit, procedure or process that affected the software development in Company A. We think this is important because it gives us the actual insight into

the habits present when Company A worked and managed the dual activity of sheet metal construction and software development.

2. **Company management responsibilities:** A theme that describes matters that were handled by the company managers (owners) that were related to risks, decision making, planning and financial matters for Company A. This one is important because it shows how the managers ran the company during the dual activity, and their reasoning as well.
3. **Beneficial staff qualities:** In this theme, any observed staff-related qualities that can possibly be a benefit for Company A's dual activity are mentioned. It can be skills, education, work experience, interests and mind-set. This is important because we gain a perspective of the staff qualities in a company with a dual activity, and they could be a contributing factor to making it possible.
4. **Staff mood:** For this theme we covered any sentiments that were expressed by the interviewees at the company, both positive and negative. This one is important because we believe this contributes to staff well-being and subsequently the long-term work environment as it shows how and what the staff (developers, managers, and even non-IT staff) were thinking during, and about, the dual activity.
5. **Comparison to other companies:** This theme holds any comment made by the interviewees that states a difference between Company A and other companies in the same industry, or in the IT industry. This is important because it will give insight as to how and what Company A does differently to other companies, and to get an understanding of the reasoning for why they wanted to do something different.

We believe that the "Company management responsibilities" theme greatly affects, and perhaps overlaps in some ways, with the theme "Software development habits", since some of the decisions that the managers made, affected directly or indirectly the development process and the workflow employed by the developers. As was mentioned in the interviews, "Going from a structured way into an ad hoc way of working, this happened because of a decision.", (as said by D1). Here we can see that a management decision drastically changed the software development methodology followed by the developers.

Another possible overlap is between the "Comparison to other companies" and "Software development habits" themes in that sometimes the interviewees compare Company A's development process to that of other companies.

The theme "Staff mood" seemed to be affected by both "Software development habits" and "Company management responsibilities" especially when it comes to the negative. For example, management inexperience in software development could cause concerns for software developers, also, the pace and cost of the software development, among other things, could cause concerns for management.

## VI. ANALYSIS AND DISCUSSION

In this section, we analyze and discuss the data that we gathered and how we view it from the perspective of being familiar with software engineering, but not so much in construction work.

We believe that all the interviewees were collaborative with our study, and therefore were not noticeably cautious when answering our questions, which we believe helped us in our study. We think this is a consequence of our attempts to loosen up any tension in the interviews by not immediately starting to ask questions but rather taking some time to introducing ourselves and by explaining the purpose of the interviews and also of our research project.

We had originally planned for the interviews to last around 30 minutes, but the duration of the interviews turned out to be quite varied. Our shortest interview lasted only 11 minutes with N1. The reason why this interview did not last longer was likely due to the fact that N1 was not involved in the software development and was not significantly affected by it, as can be seen by one of the answers: "I feel that I don't work so much with the IT part, so I don't feel that it has affected me perhaps that much." (said by N1). Perhaps we should have designed the interview questions for non-IT staff with less assumption that they would be affected by the development process and more about their work environment in general. But then again, this research must stay on topic on software engineering, and those kinds of questions would give us answers that would likely not help us in our research if the staff member were neither affected nor had any insight in Company A's development process.

One of the managers had much less involvement than the other manager in the actual software development process and was mainly involved in the decisions, but the interview still took 55 minutes. That manager was, however, affected by it in terms of managerial and financial parts. As mentioned in the Methods Used section, the manager went a bit off-topic during the interview, which might be a sign that our questions were too open, and too many for the managers. Perhaps we expected the bulk of the information to be said during the interviews with the managers, which led us to spend most of our efforts on designing the questions for the managers. A distinction between the manager that was leading the software development process and the manager that was mainly in charge of the business would have made the questions more fitting for the interviewees, and therefore would've made the answers clearer and more concise, instead of giving a long answer to make sure that they got their point made.

**Key takeaways** – Here we will present a summary of what we believe are the key takeaways from our findings that we think contributed the most to making the dual activity of software development in a construction company possible, which also contributes to answering both RQ1 and RQ2:

1. Even though the managers didn't have much knowledge in software engineering and methodologies, especially in the beginning, and it led to some issues, they were doing research about it, and were also willing to change into an agile way of working by having meetings every day and using cards on a Trello board to keep track of what was being done. They were also trying to adapt as they learned more when they went through the process.



2. Each of the people at the company we interviewed had a wide range of education and experience, some of them in multiple fields. This could suggest that the skillset in the company will likely aid the dual activity that software development brings to a non-IT company. And also, that the managers had experience in different fields and could delegate areas of responsibilities between each other.
3. Company A was careful about how they spent their money, but they were still aware that financial risks need to be taken to some extent in a project like this. We believe that being careful with how the company spends its money is a good habit.
4. It was clear that Company A struggled at times with managing software development alongside their main business, but there was always a will from the managers to continue through obstacles and an understanding that a project like this could mean it may only be beneficial later on.

**Main findings** – Next, we will go over the main findings we made, both positive and negative, after having analyzed the collected data:

From the interviews we noticed that at the beginning of the development process the managers knew very little about software development methodologies. However, as the process went on, they started working a bit more agile by implementing tools like Trello, to keep track of the work that was being done day-to-day and having meetings every day to check what had been done the previous day, what they were going to do, and what they planned to do. It therefore seems the managers were willing to try new approaches that could benefit the development process, although the process was very much ad hoc for a while after the consultant team left. This is related to RQ2 since it shows that Company A wanted to adapt to an agile way of working, which gave the management the ability to monitor the work and keep track of the progress that was being made by the developers, which gave the responsible manager ease of mind.

However, we can see that there were some faults in the communication between management and the developers, which led to some difficulties. Like mentioned, the managers did not seem to have great software development knowledge despite M1 doing research on the topic to make the development better, “We did a big research in the beginning about how and what kind of code platform we should use.” (said by M1). However, D1 had much more software engineering knowledge, but found it difficult to understand what the management was asking of them in terms of requirements, and this can be attributed to the knowledge gap between the two groups.

One important thing that we gathered from our interview with N1, is that that non-IT role seems to have been completely unaffected by the development process. This indicates that other staff members in the company may not be affected negatively by a dual activity like Company A’s. It was clear that to take on the dual activity of working with both sheet metal construction and develop software at the same time requires will to continue despite obstacles. It seemed both managers agreed to this. M1 said that Company A should strive to make changes, and that is the reason why they are in the industry, to make Company A different and better than

other companies. Continuing with this thought, M2 argues that both managers are determined and will not give up, despite very difficult situations. Also, not to be afraid of taking financial risks, which a project like this can certainly be. “It is precisely that it is a financial risk-taking to invest in something that we are now sitting on a couple of years later and do not actually use fully. So that so far, from an investment perspective, it has only been a cost more or less.” (said by M2). It seems to be that projects like this will need both financial and psychological resilience from the side of the people responsible for the company. This is related to RQ1.1 as it explains how the management needs to take the risk of investing, but not immediately have a use for the app and hence not seeing the benefits until years later.

The managers express that although they do not consider the software development to have been as successful in all ways as they had hoped for, they do believe that in time the software will be a success and the results will be beneficial to them. Also, they have lost money and time in the process, “Well, in a certain way it hasn’t been really successful to this day, it depends on which angle you look at it from. We have lost a lot of money and a lot of time.” (said by M1). This reinforces the idea that there needs to be a will, as previously mentioned, and that one must look towards the future and not stare blindly at the problems that appear along the way.

It seems that the development process was supported by the managers but there was not a lot of interest from the staff members who were the ones that were going to use the software. M2 believes that it is good if there is an urge from both the inside and outside, and that employees should be part of testing the app and come back with suggestions to the persons responsible for development, in other words, M2 believes that the development could have gone in a different direction that would have included the staff that would have actually used the software. In previous research [9] it has been found that it’s important not only that the managers get involved in IT processes, but also that the managers give the employees the motivation and the ability to contribute ideas and to further help the process, because if they don’t then they will get fewer ideas suggested. It seems that M2 understands that Company A needs not only M1 to be greatly involved in the software development process, but also that the people who will use the app should be included in the process to give feedback about the software.

It hence seems that the most weight is, and has to be, placed on the people in charge of the company in order for a project like this to be possible under these circumstances.

Most of the people we interviewed had either studied in multiple kinds of fields or had work experience in more than one field. For example, M2 had experience of taking responsibility from other jobs, which could have given them a better chance at managing the business side of the company. Meanwhile, M1 had studied a bit of web design alongside their university studies, which probably helped them with the design and to lead the software development process in Company A. Managers with various skills and experiences seem to have been a positive force in the dual activity that this company had. We believe this is a contributing factor to answering RQ1, because having multiple relevant skills is likely to be helpful when working with multiple activities, such as software development in a non-IT company.

When it comes to the work environment, there was a bit of disagreement between roles when asked about the work environment. For example, D1 did not think the company provided a good work environment during D1's time as a developer in Company A. On the contrary, D2, who was also a manager, believed that the work environment was good. Perhaps this disagreement existed because there didn't seem to be a clear and agreed definition of how a good work environment should be in the company. At least we did not find any indication of this during our interviews. This could help answer RQ2.1, since it shows how the work environment for the developers can be affected by the lack of definition of a good work environment, which may be a consequence of the fact that Company A is fundamentally not an IT company.

Last but not least, the managers of Company A mentioned several times how the app they are developing will bring value to their sheet metal construction business eventually. Both managers seemed to believe there is a digitalization gap in their industry and when comparing themselves to other similar companies, they believe they are doing something out of the ordinary that will give them advantages. For example, M1 talking about how the development of their software has affected Company A's position regarding competitors: "Profitable, a lot more profitable if you work in the right way. Growing faster, getting more customers, better quality, more environmentally friendly, a lot of things better." We think that if there is a belief that the finished software product will bring value, it will help the mindset of the staff to go through with it. If a company is thinking of developing software, it could help to make sure that what they develop has value in the main business, instead of developing software that has an unclear purpose.

**Coding findings** – Here we discuss findings related to the frequency of some of the codes in the interviews:

We found during the coding process that D1 discussed many managerial aspects in the interview. This could mean, given the software engineering knowledge that D1 had combined with the work experience, that D1 had knowledge or ideas on the best way to carry out a development process.

Another finding is that M2, who was not involved much in the software development process, had the most concerns out of all the interviewees. This could be explained by M2's little insight into the process, which may have made the manager feel more worried about how things are going or not understanding how resources are spent, or if they were spent correctly. However, M2 was also the one that expressed contentment the most out of all the interviewees, which could be explained with the same reason that they did not have much insight in the development, and it was clear that M2 viewed the development of the app as something that will pay off in the future.

The developers showed less contentment than the managers did. This might be because the managers, as said previously, viewed the development that it will provide value in the future, while the developers focused more on the process at the time, where there may have been some issues. For example, D1 believed the work environment was complex because that developer did not understand how a construction business worked, and therefore had to work harder to understand the requirements that were given to D1.

**Peculiar findings** – These are things we found hard to reason about, but we still try to give our view on it:

We noticed there were a few contradictions between how the management and developers experienced the development process. D1 argued that the development process went fast, which led to a sacrifice of quality. However, M1 argued that Company A always valued quality over coding fast. This difference in opinion seems odd, but perhaps M1 was just stating what would be the perfect scenario rather than what actually was done in Company A. What can be said, however, is that D1 thinks that quality has been put aside by such a large degree that "The operational part of it took a kick from that, maintaining that code base is going to be tricky because I'm not sure if there are any unit tests written at all." (said by D1). Perhaps quality was not the main focus during the first phases of the process, but it has become of higher value to the management of Company A now.

The managers had originally hired a consultant company to develop their software for them, but after a while they moved towards using their own developers. M1 expressed disappointment regarding the pace of the consultants considering Company A paid them a lot. M1 explains there were excuses and things didn't get finished. This heavily implies that was the reason for the change. The managers thought that that the consultant team itself was not good, but they considered that they had a good way of working, and therefore decided to replicate some of the ways of working that consultant team had and apply it to their own developers.

There also seemed to be a bit of ambiguous information in regard to how the software development was handled in Company A. For example, M2 said there were no in-house developers, whereas M1 said they started with consultants but later moved to hiring their own developers, and we also interviewed D1 who was, in fact, a developer that worked for Company A. This ambiguity can perhaps be explained by the fact that M2 was the manager that wasn't involved much in the details around the software development.

**Future work** – We believe there is good potential for future work on this topic. It should focus on widening the knowledge about non-IT companies that are, or have been, developing software as a dual activity alongside their main business. There could for example be case studies on individual companies, or surveys in multiple companies. This will give more understanding of beneficial habits in these companies, managerial aspects and how the work environment is affected by it. Also, when selecting the target population, future researchers should try to also include non-IT staff to see whether or not they feel affected by the dual activity that software development introduces. Future work on this topic could be important because more and more businesses seem to be digitalizing and introducing the use of software to help their processes.

**Limitations** – We identified some threats to validity, weaknesses and some other points worth mentioning that can be applied to our research

An external validity threat is that, considering that we are studying a small company in a field where software development does not seem to be common, there may be a question of the level of representation if comparing multiple companies of similar size, and different companies have

different needs. However, the results we find could still be considered useful as some of them could apply to other companies in similar situations that have a need to develop software.

As for weaknesses in the study, one weakness is that when working with the translations and transcriptions, there could be a challenge to get full accuracy on the correct meaning of everything that is being said. Another weakness we identified related to the company we have picked is that it is a small company with only a few employees. This makes our data sources in the company few, but we compensated for this fact by making sure our interview questions were well designed, and encouraged the interviewees to talk freely when answering, and to develop their ideas further. We also compensated for this further by finding data in literature that can help us make our conclusions more valid.

It should be noted that some confirmation bias may be present in our thematic analysis, since we took a deductive approach for some of the codes and therefore might have overlooked some information in the transcripts that we did not think was valuable. However, as described above, while going through the dataset we found some topics that we had not considered before and decided to add them to our process of analysis. This shows that despite having some ideas before beginning the analysis process, we made some changes that we had not expected.

A delay of over a week in the project was caused when one of the authors got sick. It was during the week that we planned to work on transcribing and translating, and that author could hence not work as much as he wanted to, although some work was still carried out. We were also going to try to arrange one more interview with a non-IT staff member at Company A during that week. However, due to the delay caused by the sickness, we agreed that this interview was going to be set aside as it was more important to get to work with our already collected data given the timeframe we had. The delay also caused us to have to transcribe and translate at a higher pace than we wanted.

Finally, one of the researchers in this case study is acquainted with two people in Company A that were interviewed. For this reason, we designed the interviews with those two interviewees in a way that the non-acquainted researcher was leading and asked the questions of that interview to avoid any bias. The acquainted researcher was managing the recording and could still ask follow-up questions.

## VII. CONCLUSIONS

The problem we identified is that smaller non-IT companies have more difficulties when developing their own software, because it leads to a dual activity consisting of managing their main business and the software development simultaneously. We performed a case study on a small sheet metal construction company that has been developing software as a dual activity, meaning that they had the additional tasks involved with software development alongside their construction business. We collected our data by interviewing people at the company with different roles, and then performed thematic analysis on the transcripts of the interviews.

We found that in Company A the management was the driving force in the software development, and we believe that the determination of the company managers has been, and remains, one of the most important factors in Company A's development. But we also believe that the skills and experience of both the managers and the developers have played a big role in it, in that most of them were skilled in multiple areas allowing them to take on tasks that they otherwise likely wouldn't have been able to take on. We found that the managers in Company A viewed the software development as an investment for their sheet metal construction business and understood that it wouldn't pay off until years later.

It would seem, however, that if there is a lack of agreement on how a good work environment should be, it will probably be difficult to align what the managers and developers think about it. This could lead to managers falsely believing that the developers have a good work environment, and that they don't see that things need to change.

Another finding is that a manager of a company like this will likely not have significant software engineering knowledge, which can lead to managers giving vague requirements to the developers, which, in turn, can lead to difficulties in the development process. Despite this lack of knowledge, the management implemented parts of an agile way of working which became beneficial and helped Company A.

This topic holds opportunity for future work by studying more non-IT companies that also develop software. This will widen the knowledge on how these kinds of companies handle projects like this as there will likely be an increased need for digitalization in businesses.

## ACKNOWLEDGMENT

This paper was written in collaboration between both authors who contributed to it equally. We would like to thank the company managers and employees at Company A for participating in our case study and letting us interview them.

## REFERENCES

- [1] "Shaping the future of construction a breakthrough in mindset and technology," World Economic Forum, 2016.
- [2] M. Kießling, H. Wilke and L. M. Kolbe, "Overcoming challenges for managing IT innovations in non-IT companies," presented at the 16th Americas Conference on Information Systems, AMCIS 2010, Lima, Peru, Aug. 12-15, 2010, pp. 1-9.
- [3] C. Ponsard and J. -C. Deprez, "Helping SMEs to Better Develop Software: Experience Report and Challenges Ahead," presented at the 2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), Gothenburg, Sweden, May 30-June 01, 2018, pp. 213-214.
- [4] M. Mandviwalla, R. Flanagan, "Small business digital information in the context of the pandemic", *European Journal of Information Systems*, vol. 30, no. 4, pp. 359-375, Mar. 2021, DOI: <https://doi-org.ezproxy.ub.gu.se/10.1080/0960085X.2021.1891004>.
- [5] P. Runeson, M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, 14, pp. 131-164, Dec. 2008, DOI: <https://doi-org.ezproxy.ub.gu.se/10.1007/s10664-008-9102-8>.
- [6] S. E. Hove and B. Anda, "Experiences from conducting semi-structured interviews in empirical software engineering research," 11th IEEE International Software Metrics Symposium (METRICS'05), Como, Italy, 2005, pp. 1-10, DOI: 10.1109/METRICS.2005.24.
- [7] M. E. Kiger and L. Varpio, "Thematic analysis of qualitative data: AMEE Guide No. 131," *Medical Teacher*, vol. 42, no. 8, pp. 846-

854, May 2020, DOI: <https://doi-org.ezproxy.ub.gu.se/10.1080/0142159X.2020.1755030>

[8] J. Saldaña, *The Coding Manual for Qualitative Researchers*, 2nd Ed. London, England: SAGE Publications Ltd, 2013, pp. 3-205.

[9] M. Kiessling, H. Wilke and L. M. Kolbe, "An Organizational Model for Managing IT Innovations in Non-IT Companies," presented

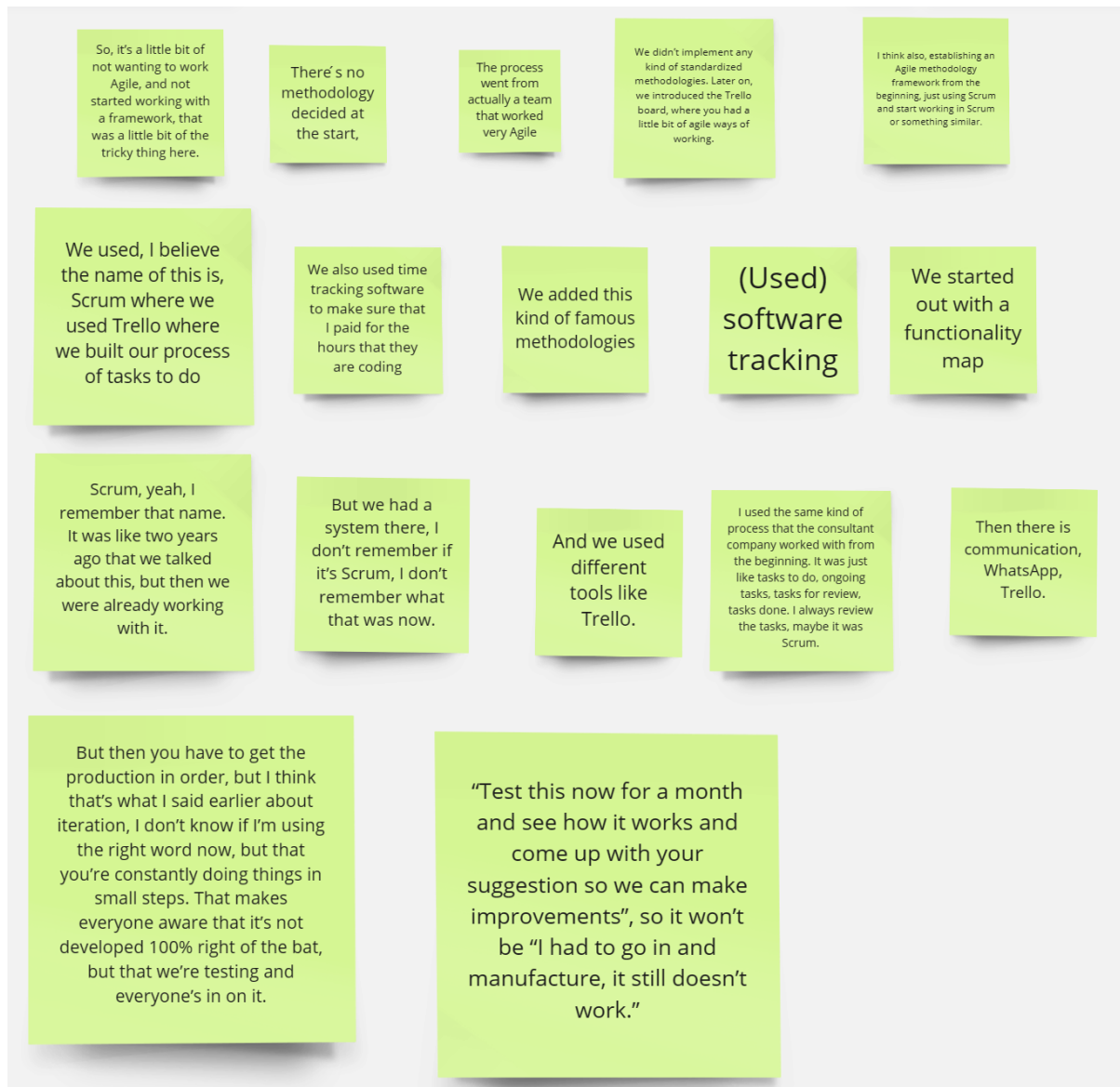
at the 2011 44th Hawaii International Conference on System Sciences, Kauai, HI, USA, Jan. 04-07, 2011, pp. 1-10.

#### APPENDIX

Codes in the online whiteboard

## Appendix – Codes in the online whiteboard

### A Workflow methods and tools







## D Financial issues





# E Concerns

but nobody actually knows what I know and it's hard for them to understand what I mean

And it's hard for me to understand what they mean because they come with a different set of experience and another kind of vocabulary

It's hard to go to the same alignment, so that we're all on the same page. That's a bit of a tricky thing in the beginning.

It's hard to change the way of thinking of someone else. This is how it should work, and it's hard to tell that when you can't really point to that. This is going to be so much easier if you can't do this.

There was a bit of conflict

(about successful software development) it hasn't been really successful to this day

(their app) if we decide today to not use it then it's been a lot of wasted time

There was some lag sometimes when I used some other foreign consultants

I don't have the eye for looking at code, of what is good quality and bad quality of writing code lines. So that is the biggest risk here

If you want to do any changes, it's going to be quite complex to understand what, where and how you do that change. That's why, I think, it would not be sustainable in the long run.

It was not the amount of work that piled up, it was mostly the uncertainty of the requirements.

The uncertainty of what needs to be done was sometimes quite frustrating because, as a developer, implemented something and then it's not what was expected, or it was not exactly what was expected. It was not clear.

When you are already done, but not ready to because the software engineers were not provided the color codes, how many pixels big the font should be, etc.

Yeah, there was some period a few years ago, but in most cases I worked during my studies.

So, I believe that was the slow progress of actually getting the information from my guys here, and add it to the app

(Worry about something not succeeding) Yeah, for sure, always

There are so many people that give up easily

the implementation of the app has not worked fully for various reasons

It was a lot of hassle in the beginning

It was hard to align the mindsets between the software development and management in understanding what the software engineers would need from the management, to be able to do the job, what kind of specifications.

But if the management already knew this, why couldn't we get this from the start? Because it's a very tedious process of building ad hoc.

But this was done with disregard for not having a process established, as a methodology, on how to deal with changes and so on.

The operational part of it took a kick from that, maintaining that code base is going to be tricky because I'm not sure if there are any unit tests written at all.

(About their app) We may not have had the time to spend on it, which has been a big minus

We've been too few people who have been at the core of the app development

(About the consultants) They probably didn't have the knowledge to take the next step, it probably wasn't the right company, I think

Their entire team was maybe not top, it was maybe one in their team who had the total knowledge and maybe understood

(About the consultants) Then there is the flip side of the coin that they may not have had time, they may not have had the required knowledge all the time

So, it's a lot of quality, in terms of software, that has been the cost here, for the sake of moving fast.

the other programmers that were programming didn't fix it

don't get screwed by foreign consultants

it's easy that someone is building something wrong

it was a little bit bad for me sometimes, I was stuck with things

(Regarding distribution of work between developers) I always used one, because if there were more, they would be bad

It's hard to monitor three or four programmers and it gets chaotic, it's big chaos

Sometimes I was like "Oh my god, I have to design these things and do this."

Sometimes I felt pressure when I had a really good coder, "damn", I need this dude, I don't want him to go away now."

My guys here didn't supply me with all necessary information

it has been a bit of a bumpy road to get to where we are. It has taken much, much longer

Using consultants may also have been, perhaps not entirely optimal in all situations

if that guy barely puts work into it and the others on their team are kind of half-motivated, what will the result be?

There have been shortcomings in our team as well, in that we have not been able to develop on the production side itself. Different models are on, it has been a huge miscommunication.

So, I would probably say that we haven't really been a one hundred percent team to pull it, quite simply, all the way.

The easy stuff is not done enough

In the beginning there was stress

the stress was more that I didn't get the product information

(Analogy) I was fixing the garden, I was fixing the garage, I was fixing the flowers, but the freaking kitchen never came.

It made me disappointed and stressed a little bit

The team around it may not have been optimal or one hundred percent in any case

The will has been there but has been stopped by one thing and then the other.

Manager about his knowledge in software engineering) it's possible that not understanding the complexity had had a negative effect.

The business itself had taken much, much more time in the hours of the day, so you have not been able to put time into this

I think we on the production side would have been somewhere else if we weren't stuck with projects that have been what they have been.

we don't want people to do bad things

Unfortunately, there were some changes during the time of the code. Oh, we didn't think about this kind of thing the beginning

(About staff on the question of more software solutions are needed) There are many ways of doing things, but it's difficult to choose the right one. That's the biggest challenge, when there are many ways to do something that you need.

(Software solutions for non-IT staff) it can take a while to find all of them, before you build it up to what you need

(About their app) needing to be customized to their business) it's so many steps, it's so complicated stuff

even if it didn't work fully in any way

The app has not been given a fair chance

I feel it would be a freaking mockery if we did not use the app

The first hurdle that was tricky to solve, was to understand the user needs, the user requirements and the business requirements for the company to be able to provide.

(Not using the app fully) It's a bit of a shame on us, or a lot actually

I started to be more interested about the code and got frustrated about things when it was going slow

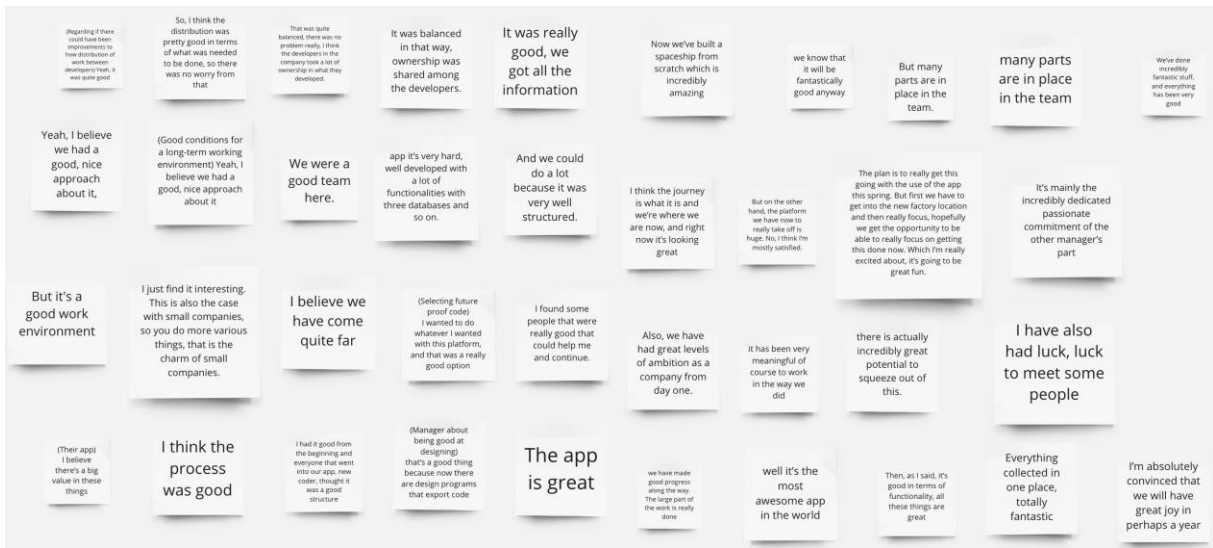
In the beginning we used a simulation system and then when the simulation started because they were so slow, and they didn't get things done, even though I paid quite a bit

(about the consultants) I felt like they didn't get things done and there were excuses and things like that

That thing would probably kill most of the companies, when you get wrong consultants

(regarding using a code we will see in the future) I didn't want to be locked and stopped

# F Contentment



## G Beneficial staff qualities

Merging software engineering and construction, because they're two different worlds, two different experiences and so on. So, it's a little bit of understanding each other and so on

Communication skills

That the people here, that we're working with on this app, really had the will to make it.

Making sure that we believe in this, and not giving up.

Drive and interest from both managers, so from both owners.

But it's the drive and interest, I think.

To be interested in solving the problems. It comes to the main people that "Ok, if you have the will to do something, do it! And learn to do it"

Not only want, but that there is a person who has the time to be able to sit with it, and simply invest time

they helped by saying which code we could use and what kind of code we are going to see in future, 5-10 years.

Yeah, I was a little bit more of a technical decision lead, I was usually the one who discussed the technical solutions and so on

I studied three years in something called Information Architect, that has a little bit of both IT design, IT management, and IT development, at Bachelor level

I'm tech-lead in a team

And also having good coders.

Designing things, branding things, I was studying that too, by myself

And I have to keep the good developers here

I went very much from designing front-end stuff into actually developing a lot of backend and cloud-computing and so on

I've been doing a lot of front-end stuff between 2016 and 2019, and then from 2019 until today it's been cloud-computing and so on

Professional experience is three to four years

(Manager experience) But then when I worked as a chef, I had a lot of chef responsibilities

better programmers

Getting help with picking the right platform from someone who understands, and who knows the business.

I actually have three degrees.

I like to do the designing of websites, and doing that, so I was interested in HTML and CSS basically.

I am a real-estate agent and I have a civil economist degree and a civil engineering degree.

I worked there partly as a consultant and then even as a manager.

I worked as a construction project manager

Describe my experience. It's probably a concoction of what I've learned at school and what I have an interest in, and also what I've worked with previously.

Exactly five years in this role, but otherwise for ten years.

(Manager background) I've been a chef, I'm also a trained sheet metal worker

(Manager experience as manager) I have those six years as a leader

Today I know a lot, but under this process I have learned a lot of things and been involved and can catch up on information quite fast

it depends on how good you are as a person to write notes that "I fixed this thing here"

It's a personality thing, if you are structured as a person and want to make the world do better or make the code do better

I had one person that was a really good coder, and he helped me during the gap between the Swedish consultant company and helped me to get further on and continue. He was the guru, pro-coder

(Manager) But I know how to do design

(Manager education) I have studied the first two years, engineering subjects to a large extent, mathematics

(Manager) I'm good at designing stuff

(Manager experience as manager) There is a lot of management literature and I have read an incredible amount of it

I know that I am a good designer and manager

(Manager education) Masters in Industrial Engineering, business administration and engineering, mixed together

I was done with my masters of innovation and management

(Manager education) So, if you look at it purely academically, it's probably high school qualifications and scattered university courses

During my studies I learned a lot about software. I wanted to do startups when I was in the university, that's why I learned WordPress and HTML

that the other manager has the training and the skills that he has had

(Manager experience as manager) Six years

both me and the other manager never give up. For better or worse

The other manager has a completely different knowledge now to what he had when he started this as well

(Manager experience as manager) I have done a lot of manual work, but then you have the work that comes with staff and stuff like that

## H Comparison to other companies

(Construction and metal sheet industry) Those kinds of companies are a piece of a business sector that is not digitalized.

I would compare the company with other startups, I would not compare them with the huge enterprises.

It's a little bit on demand, I would see as the difference to other startups.

If you look at other IT startups, of course those would have more of an IT methodology and so on.

If you look at another kind of startups, when they are doing an IT solution, my guess is that they would more outsource that kind of thing to a completely different team. Where a consultant agency would have been able to provide the whole, build and maintain, the complete solution.

If there is a startup that doesn't have the competence in their business, they would reach out to a consultant agency and say that they need to have the design, and so on, for this app that they want to build. And then this consultant agency puts together a team that does all of this together as a package deal for the startup.

I think that would be also a difference from how the company is working. The company is even more like "We want this extra page to be added. Ok, let's find someone who can add it". It's even lower in the details of building, in regards with software engineering.

if you have a funded startup, it's easier

there was some competitor out there, but they were far away from doing so pro as we do it, because they had the belief that "Ok, it should be possible to do this."

One day you do something very concrete, while the next day you do something completely different, so it's sort of small company nature.

Sometimes there have been finished programs, or finished systems, and then you have to adapt them a little. But then at some company, I was involved in building it from the ground up, so "What are the things we need?". And it takes time.

Wanting to do something different from what all the other sheet-metal companies are doing.

Which is about the same work I do now but at a slightly larger company, where there were several project managers who did pretty much the same thing.

It was to be the next generation sheet metal construction company, so that means that we want to change and make the industry better

So, the package we have, made out of all these things that are more technical, that we're using compared to our competitors

The idea of me jumping into this industry is that we have to be something else, we have to be better, we have to strive to make some changes

I believe that the guys that work here for us know that we are the future of this industry

Profitable, a lot more profitable if you work in the right way. Growing faster, getting more customers, better quality, more environmentally friendly, a lot of things better.

I'll probably have the most fancy office in our industry

But it has led to so many other things, marketing, that we are doing it has really given a bit of PR as well. That you are still doing something different than what your competitors are doing.

The consequences of that would have been that it had been a classic sheet metal company. It would have stopped there. Then you haven't had any increase in production. Then nothing has been made more efficient. So that's it. It would have been an ordinary traditional sheet metal company.

We have had the opportunity to take jobs that you normally wouldn't have had the opportunity to take, which would have meant that, well, then you would have been a traditional sheet metal company and then you might not have generated the profits to be able to do other things and so on.

We have developed our own website not through a template or something, like raw code.