Stockholm School of Economics
Master of Science in Economics and Business
Thesis within Management, 15 ECTS
September 2, 2013

Managing in an Organized Anarchy

An Autoethnography about the Struggle to Deliver in a Young Technology Company

Revised Edition

Author Anders Sjöqvist (20487)
Academic Supervisor Lars-Olof Lychnell
Opponent Lena Lindvall (20457)

Abstract

An organized anarchy is governed by vague ideals, and all employees help to define the strategy. Solutions are created without knowledge of the problems and many of the solutions get rejected at an early stage and end up in the "garbage can." The garbage can is revisited when new problems surface, as the contents might contain a solution to one of them. Such a structure often works well for organizations like universities. But it should be noted that while the organization is good at discovering new possibilities, the problem-solving speed is lagging behind.

This thesis describes and analyzes an organized anarchy from the inside, through the eyes of a middle manager. The organization in the study was under pressure to deliver a product, but was better at coming up with creative ideas. As the pressure increased, the lack of accountability made the teams search for people holding the sole responsibility for the setbacks. The dominance of short-term goals limited the focus on sustainability and a healthy work environment. As a root cause, the author identifies the organized anarchy—further reinforced by the results of scapegoating and negative leadership—and not the individuals themselves. While individuals are certainly responsible for their own actions, their range of options is limited by what the organization permits and encourages.

The thesis suggests that certain organizations are worse at delivering value than others, despite the competence of their employees, simply because of their structure. The topic is important, as huge amounts of resources are wasted while the organization is suffering an internal crisis. The use of autoethnography as a method helps employees and middle managers to identify with the author and classify the organization they're working in, and it helps business leaders to understand how their choices affect the working environment for their employees. Also, the thesis can help in diagnosing an organization that has problems delivering.

Keywords:
analytic autoethnography, organized anarchy, garbage can model, leadership, corporate scapegoating

Acknowledgments

I'd like to thank my academic supervisor Lars-Olof Lychnell for his continued support and patience during the entire research process. He was the one who originally suggested using a self-narrative method, which is rather unconventional within management studies, not expecting me to go along with it. Yet, despite being new to this category of methods, he stood by me and helped me find the answers to my questions.

I'm also grateful to Åke Freij for taking time from his research to help me out, not only with understanding the autoethnographic method but also by discussing which of my experiences would be most interesting to a reader.

Furthermore, I'd like to thank my parents—Arne and Maud Sjöqvist—for their support and their help with proofreading, and finally those of my former colleagues with whom I was discussing the events during my time at the company as well as afterwards, and who later volunteered to read the thesis as it was getting closer to completion. For the sake of anonymity of the organization I made the decision not to mention any names, but I want to tell them this: You helped me to broaden my perspective and to better understand what was going on. Without you, I couldn't have found the answers that I did. When I shared the draft with you I expected suggestions for alterations but received none, which I'd like to think means that I covered all that was important. But your enthusiasm for what you read was even more valuable to me, as it taught me that this is not just a story about me. You told me that the thesis helped you to reflect and comprehend your own situations. Each one of you has unique experiences from your time at the company, but all your experiences fit together. When I became too deeply involved with theories, there were moments when I feared that my ideas had lost touch with reality, but you helped me understand that despite the subjective nature of the research method, the results are quite objective and likely relevant in many other organizations as well.

Contents

1 Introduction

Should management for a start-up really have to care about building a culture and detailing areas of responsibility, requirements for communication and which deliverables to use for measuring performance? It seems so boring and bureaucratic. Shouldn't they be allowed to assume that everyone's working towards the same goal and for the good of the company? The organization should be assumed to work efficiently as long as all the employees are good enough, shouldn't it?

This thesis will tell the story of an organization, with a huge potential because of a skilled (albeit young and inexperienced) staff and powerful investors, that has had to struggle constantly because of difficulties to deliver on the promises to the shareholders. It will then use theory to describe how such an organization can develop ideas and reject them before they're even tested, and how it can cause an atmosphere that will embrace sycophants[1] while driving talents away. Although the organization was trying to pinpoint its problems and solve them, it's important to remember that solutions which point to any one particular employee or process are likely wrong. The organization should instead look for causes among their policies, to find out how they could have allowed it to happen.

Of course, management is not simple. Some companies flourish while others fade away and die. Sure, this can be due to a business idea that simply wouldn't cut it, or economic development that renders some services obsolete. But when a company has to struggle simply because of its internal culture and structure, then the economy and society are worse off as a result of it. When a company is working for years to develop their product without ever reaching the finish line, then huge amounts of money are lost as development costs and missed opportunity. If the company ultimately fails it might not be because the idea itself was flawed, as it didn't even reach the market where it would either sink or swim on its own.

The topic about problems to deliver is an important one. The organization in the study wasn't

[1] flatterers, bootlickers, yes-men

a small bootstrapped[2] venture run from a basement. They had a few dozen employees and offices in both Beijing and Shanghai. They attracted foreign investment, and the most prominent investor was a venture capitalist that had invested in several renowned IT companies. Why couldn't they spot these problems? They clearly invested in the idea that the company had presented, but shouldn't they be able to guide the company onto the right track also in terms of processes?

Especially as a start-up, it's important to understand that problems might arise when the company is growing and responsibility and accountability scatter. In 2013, this company was still presenting itself as a start-up, even after having been around since 2007. The will to stay a start-up might be a sign that they desired the enthusiasm of the founders in a new company, but then got the disorganized way of working along with it.

The method of the thesis is unusual, as it presents a narrative from the perspective of a middle manager within the organization. The narrative is based on memories, diary entries and discussions with peers, but not on any formal interviews or documents. The contributions of the thesis are partly the description of a problem, the set of theories used to analyze it and the conclusions, but also the story itself and this method for writing an academic text.

After the introduction, the methodology of analytic autoethnography is presented in section 2, followed by a theoretical framework for describing problems on an organizational, group and individual level in section 3. The largest part of the thesis will be the story itself, presented in section 4. The story is analyzed in section 5, which is followed by conclusions in section 6 and a discussion of the results in a bigger perspective in section 7.

1.1 Research aim and research question

The aim of this thesis is to investigate and analyze a concrete example of how a small multicultural company deals with internal conflicts and its own identity, and how this reflects on its ability to deliver. The ambition is to provide a case study, but with focus on my own role in the company and how I interpret events that I am an active participant in. This will then be complemented by a theoretical framework, to help entrepreneurs and other decision makers build solid structures and deal with potential issues within their own organizations.

Having been a part of a rapidly changing organization which was struggling to build a successful team but faced persistent problems, I can describe these processes from my own perspective rather than as an external observer. Although some of the attempts the company made to improve the conditions turned out to be failures, something good can come of them by learning from the mistakes and helping others in similar situations. My desire is to pass on these experiences, with my subjective observations in focus.

From an external point of view, it would be possible to examine how the company deals with communication, which method for software development it uses and so on. Many such attempts to find causes for problems had in fact been initiated by the company, but my ambition is to find a theory that transcends all the others and affects how people think. To do that, I want to use my own experiences to figure out people's motivations, since formal interviews might only give the answers that the interviewee either thinks are prudent, or are quotes from the official presentation of how the company operates, regardless of whether they're accurate or not. By conducting research on my own, it might be possible to dig deeper than that.

When analyzing conflicts, especially those that one is a part of, it's easy to find oneself in situations of finger-pointing and directing the blame for the conflicts at others. I believe that this is inevitable, even in this kind of thesis, not only because parts of one's personality is visible only to others and the reasoning of others will always be more difficult to comprehend than one's own thoughts, but also because this thesis aims to give a personal account of the events, giving no promise whatsoever about objectivity. However,

[2] to make it on one's own without external help, for example by reinvesting earlier revenue; from the expression "to pull oneself up by one's bootstraps"

it is important to remember that the thoughts and feelings described are keys to understanding the decisions and outcomes. Still, if the text is allowed to remain at that level, then the reader will have little to learn from it beyond the author's personal feelings towards his colleagues. Hence, although an event is at first described from a profoundly subjective view, the aim is to raise the topic to a higher level of abstraction and generalization.

While not an immediate goal in itself but merely a consequence of the research method chosen, the thesis serves as an example of how analytic autoethnography (see Anderson 2006a) can be used in a business thesis. Given that autoethnography is uncommon as the choice for a master thesis, the reader is not assumed to be familiar with it. Consequently, emphasis will be put on an explanation of the method in sections 2.1 and 2.2.

The research question that I'm attempting to answer is:

> How can theories of culture and teamwork explain irregularities in productivity within a young company with a largely inexperienced staff?

1.2 List of terms

This section contains explanations of some of the words and phrases used in the thesis. Although the aim has been to use as few complicated or unfamiliar phrases as possible, there are situations where not using the proper terms actually makes the text more confusing.

Agile Software Development One common way of planning software development projects, called the *Waterfall model*, is to write a document describing the desired outcome, down to where the buttons should be placed. It's possible to imagine how such a method could be suitable for government agencies, which can't sign a deal except after public procurement, where many are able to bid on the contract. Nevertheless, it makes many assumptions about how the process is conducted, as even the slightest change is normally associated with huge costs and a renegotiation of the contract. It assumes that the software can be designed as a concept without being tested in the process. It assumes that requirements don't change over the course of the project, and that the buyer will have enough money at the end of the project to pay for the costs.

In reality, the Waterfall model often turns out to be impractical, and it is often contrasted by *agile software development*, which is really a group of methods. The most famous of these is *Scrum*, where development takes place during *sprints*, which typically last for two to four weeks. After each sprint, the product is compiled into a working version that can be tested by the customer. The customer could potentially abort the project at that point, if they are satisfied or run out of money, or they could reprioritize in the list of remaining tasks, from which the development team picks a number of tasks from the top, depending on how long time they predict that the tasks will take. The person who leads the Scrum team within the company is called Scrum Master, while the one representing the customer is called Product Owner.

Another agile method is *Kanban*, which doesn't have the concept of sprints and tasks that the team is working with together. Instead, tasks are small enough to be handled by individual developers. While Scrum limits work by the amount possible within a sprint, Kanban limits it by limiting the number of tasks in progress at once. Instead of focusing on delivering a finished product every few weeks, Kanban thus focuses on getting work done by spotting and dealing with bottlenecks.

Android *Android* is Google's operating system for smartphones and tablet computers.

Application Programming Interface (API) Most computer software will at some point communicate with other software, for example when getting information from a database, displaying text on the screen or uploading pictures to a server on the Internet. In order to achieve this communication, some rules must be set up that dictate how and what kind of data will be sent in the request, and how the response is formatted. These rules are called the *Application Programming*

Interface or API. A simplified way of expressing it, is that every time two teams of developers need to make their code work together, they have to come up with an API. The API should preferably be easy to understand and to use, but at the same time powerful enough to handle complicated requests. Also, it should change as little as possible between version, to avoid breaking things that were already working.

Back-end *Back-end* at a software company is usually in charge of developing and maintaining the software that's running on the company's servers. For example, many companies have large databases of users and user-related or user-generated information. These data must be handled so that they're searchable and possible to deliver in a quick, reliable and secure way. The problems in back-end are for example about how to locate information from terabytes or more of data spread out across many servers, and tell which information the user has access to, based on complex sets of access rules.

Front-end *Front-end* has the responsibility of displaying data provided by back-end to the user through an appealing user interface. In my company, different front-end teams were working on web, iOS and Android apps, although the Android project started during my time at the company and the other ones had been running for a while. Typical front-end problems are how to create graphics and animations that will look good on a variety of devices, and how to constantly keep up with the latest updates on the servers.

iOS *iOS* is Apple's operating system for the mobile devices iPhone, iPod Touch and iPad.

Operations The main responsibility of *Operations* is the maintenance of the company's development and production servers.

Quality assurance *Quality assurance* is the activity of ensuring that the requirements are met by the product before deployment.

Royal Institute of Technology (KTH) The *Royal Institute of Technology* is the largest institute for higher education in technology in Scandinavia. I've studied Computer Science there, as well as the CEO, the interim CTO and several others in the company.

Stockholm School of Economics (SSE) The *Stockholm School of Economics* is the leading business school in Sweden, and one of the leading in Europe. It's the school where this thesis is presented, and apart from me also for example the chairman of the board and the CBO (Chief Business Officer) had studied there.

2 Methodology

As discussed in subsection 1.1, the thesis will try to find answers that are not easily identifiable through interviews. Instead, I chose a method that would allow me to unobtrusively collect data about people's behavior, to find underlying causes that they themselves might have been unaware of. "Traditional" research methods might otherwise have led me to point out problems with specific people or in specific processes, since this is what my colleagues often discussed.

I'm discussing naming conventions on page 7 in subsection 2.2, but the truth is that these self-narrative methods often vary more between specific authors claiming to use the same method than between each other when comparing their official definitions. Hence, I chose analytic autoethnography because it describes what I'm trying to achieve, not because it was the only self-narrative method available.

This section starts off with a general discussion about autoethnography, before moving on to the subtype analytic autoethnography. Then, I describe how I collected my data and how I went about describing my experiences. Lastly, I explain what ethical considerations I've had during my research.

2.1 Autoethnography in general

> "What is autoethnography?" you might ask. My brief answer: research, writing, story, and method that connect the autobiographical and personal to the cultural, social, and political. Autoethnographic forms feature concrete action, emotion, embodiment, self-consciousness, and introspection portrayed in dialogue, scenes, characterization, and plot. Thus, autoethnography claims the conventions of literary writing. (Ellis 2004, p. xix)

Ellingson and Ellis (2008, pp. 449–450) say that autoethnography presumes that the reality is socially constructed[3], and are thus indirectly saying that it's not possible to be an autoethnographer while clinging to the ideas of the Enlightenment of objectively observing, analyzing and searching for the one and only answer. Without getting stuck in the irony of such a statement, we can draw the conclusion that autoethnography is about finding *one* truth.

In a way similar to autobiography, where the life of the author is described through the author's own voice, an autoethnography describes a culture, which the author belongs to, through the author's own voice. The process of researching the self is called *introspection*, and by allowing oneself to become both researcher and researched, the author's experiences and emotions become part of the story and can be used to better understand the reasons and purposes of the author's actions. (Ellingson and Ellis 2008, pp. 450–452) Furthermore, Ellingson and Ellis (2008, pp. 452–453) even refute the classical view of objective science, by pointing out that science itself is a social construct. Subjective accounts shouldn't be seen as less rational than objective ones, since "reading emotions of self and others often forms a necessary precursor for rational action."

Autoethnography thus allows us to see the world through someone else's eyes. This could either teach us how we would interpret a situation which we haven't experienced yet, or show us how others who are different from us interpret something that we have already experienced. Moreover, it allows us to use thoughts and emotions as tools for understanding concepts that might for some reason or other be difficult to investigate or comprehend objectively.

Ellis (2004, pp. 135–137) expands the uses for autoethnography to also include self-therapy, and is paraphrasing a conversation she had with her students about this topic. This is, however, where I personally think that the method is starting to get out of hand. I believe that the strongest need for therapy is during phases when it's extremely difficult to maintain a decent level of objectivity. The author is likely to write down more than what is relevant to just convey the story, and the descriptions of other people's actions might be affected to the point where it turns unethical, as will be discussed further in subsection 2.5. Ellis (1993) gives an example from her own life, by telling the extremely tragic story of her younger brother who died on the Air Florida Flight 90 crash in January 1982, on his way to visit her. Ellis herself explains how difficult this is for her:

> Each writing and reading of my text has permitted me to relive my brother's death from an aesthetic distance . . . , a place that allows me to experience the experience but with the awareness that I am not actually again *in* this situation, and thus I muster the courage to continue grieving. This process may not be attractive to everyone. Often it was painful for me, so painful that even now in 1992, ten years later, I experience intense emotions of loss when I read this story. Indeed, my emotional reactions over the years have been so pronounced that at times I have noted them as I edited. (Ellis 1993, p. 727)

Ellis' article was published in the end of 1993, almost 12 years after her brother's death. At that point in time, it was probably possible for her to think rationally enough about it to know whether she actually wanted the article to be published. But her own description of her feelings makes it possible to doubt it. I believe that the more severe the experience was, the more thought has to be

[3] *Social constructionism* and *social constructivism* trace back to the work of Berger and Luckmann (1966), who claimed that the terms "reality" and "knowledge" are products of the society we live in, and that even the simplest common sense knowledge is the result of social interactions. This thesis will not attempt to explain this theory in any further detail.

put into whether to actually publish the text. It's likely to work for some of the people, some of the time. But I'm skeptical towards therapeutic autoethnography as a general piece of advice.

2.2 Analytic autoethnography

Autoethnographies come in two flavors—*evocative* and *analytic*. This distinction was originally suggested by Anderson (2006a):

> The author proposes the term *analytic autoethnography* to refer to research in which the researcher is (1) a full member in the research group or setting, (2) visible as such a member in published texts, and (3) committed to developing theoretical understandings of broader social phenomena.

Anderson had noticed that autoethnography was underrepresented within the genre of analytic ethnography, and came up with five key features of an analytic autoethnography:

1. *Complete member researcher status (CMR):* The researcher must be a full-time member of the group, but different from the others in that she has dual responsibilities. Apart from devoting full attention to the activities within the group, she must continuously spend time and effort recording the events. This forced reflection will also make her more attentive to the underlying processes, which of course affects her understanding of the events. Anderson also explains the difference between "opportunistic" and "convert" CMRs. In short, an opportunistic CMR is part of the group before making the decision to write about it, and a convert CMR initially has a research interest and immerses herself as a member of the group during the course of the research.

2. *Analytic reflexivity:* In traditional ethnography, the researcher limits the focus on herself, and instead directs it outwards. In the cases where there is a reflexive part in the text, it is clearly separated from the analysis of the external culture. In analytic autoethnography, on the other hand, it's vital for the researcher to incorporate a reasoning around her own beliefs, thoughts and actions to emphasize on the awareness of the effects of the researcher on the group and vice versa.

3. *Narrative visibility of the researcher's self:* The researcher should not only reflect on her thoughts and actions, but also describe herself as an active participant, to avoid seeming like a silent observer. The potential pitfall here is to write "author saturated texts."

4. *Dialogue with informants beyond the self:* While evocative autoethnography focuses on introspection, analytic autoethnography needs to ensure that the results are generic enough by involving others within the group. If too much effort is spent on the self, then the genre might have closer ties to autobiography.

5. *Commitment to theoretical analysis:* An evocative autoethnography will often leave the reader after having delivered the "story," but an analytic autoethnography needs to keep going, to pierce the surface and at least attempt to analyze or draw conclusions in a broader perspective. Anderson explains it best:

> The purpose of analytic ethnography is not simply to document personal experience, to provide an "insider's perspective," or to evoke emotional resonance with the reader. Rather, the defining characteristic of analytic social science is to use empirical data to gain insight into some broader set of social phenomena than those provided by the data themselves.

One of the most prominent supporters of evocative autoethnography is Carolyn Ellis. In the same 2006 issue of *Journal of Contemporary Ethnography* where Anderson published his proposal, Ellis and Bouchner (2006, p. 432) questioned even the nature of what Anderson suggested:

> "I'm having doubts. The more I looked closely at what Leon [Anderson] was proposing, the more I thought to myself, 'but this isn't autoethnography. Why does he want to

> call this autoethnography? It's just another genre of realist ethnography.'"

Anderson (2006b, p. 455) responded by criticizing the circular argument that since autoethnography had been redefined to mean evocative autoethnography, no such thing as analytic autoethnography can exist. He doesn't recognize the right of Ellis and Bouchner to redefine the word, and claims to have defined it with a meaning closer to the original intent.

Two years later, Ellingson and Ellis (2008, p. 445) acknowledge the existence of the two separate orientations within autoethnography in the beginning of their chapter, but towards the end Ellis once again gives voice to her concerns that Anderson's attempts at widening the concept of autoethnography will make her and others like her lose control over the genre to realist ethnographers, while the name itself gets watered down until it no longer challenges mainstream ethnographies. (Ellingson and Ellis 2008, p. 460)

For my part, I find it sad that the debate must focus so much on naming conventions. I don't mind the variety of names for similar genres, including but not limited to *postmodern anthropology* (see e.g. Reed-Danahay 1997, p. 17), *postmodern ethnography*, *ethnic autobiography*, *autobiographical ethnography* (see e.g. Reed-Danahay 1997, p. 2), *autobiographical sociology*, *auto-anthropology* and *self-narrative research* (see e.g. Anderson 2006a, p. 373). In many cases, I believe that these overlap, but it should also be possible to pick a name that not only suits the actual content of the text but also encapsulates its purpose. I have nothing against Ellis or her work—in fact, I think many have a lot to learn from her way of writing—but I would at the same time love it if I could gain something more than an emotion from reading her texts.

I like it that autoethnography allows me to tell the story about who I was, how I felt and what I did. I also like it that traditional research lets me analyze, get an overall picture and convey it to others. Fortunately, it seems like analytic autoethnography can combine the best of both worlds. It would be a shame if naming conventions would discourage researchers from choosing this method.

2.3 Data collection

While working at the company, I started writing a more traditional thesis and collected data through interviews, but I had to abort this project as the workload increased and since I couldn't collect the weekly feedback that I needed. It should be noted, that any information collected during my first attempt to write a thesis for the company was discarded as the work started with the new one. Although I had collected information that the company volunteered through interviews, that information was meant to be used in another thesis and I found it unethical to make use of it once I had decided to change topics.

Having realized that I needed a new topic for my thesis, I started keeping a diary of things that happened around me. I took notes on an almost daily basis from the end of November 2012 until the beginning of June 2013, when I left the company. After that, I occasionally continued taking notes, when I was in contact with the company. All in all, I have 203 diary entries, averaging about 100 words per entry.

Apart from the diary, I also maintained a file with random thoughts that had crossed my mind about the situation of myself and the company, and I saved some of the more important e-mail messages and chat logs for reference.

The process of collecting data in this way first became a habit, and then almost an obsession. I felt like the day had been more productive if I had summarized and reflected on the events that transpired. The summaries also turned out to be a bit therapeutic. If I was upset or annoyed, I could just write down my thoughts and then be done with it. The thoughts was already there, written down and saved for future, so why would I have to spend more energy on them? Furthermore, I found it rewarding to be forced to write down a summary, where I needed to reflect not only on others' behavior, but also on my own. Actually, most of the material for this thesis is the result of the thought process itself, associated with writing notes, and not the diary. By writing things down in an organized way, I could also keep them organized in my mind. Although the diary became my most important tool to develop an understanding

of the culture and events, I never really felt that I needed to read it in the end.

The information I collected has never been shown to anyone else—I consider it too personal to share with others—but I have often discussed things with my colleagues, usually not with the intention of collecting information but simply because many of us found ourselves in frustrating situations and needed to talk about it. My perception of things is of course to a large extent the outcome of these conversations.

2.4 The process of writing

As I first started writing down my thoughts about my experiences in the first draft of this thesis, I wanted to write down everything that came to mind, to let my ideas and memories start to flow. I knew that later on I'd have to filter these stories with respect to the theories that I would eventually select.

Once I had started writing, I kept writing until I had 17 pages of stories off the top of my head, only occasionally referring to my notes to verify some details or order of events. With a few stories still on my mind, but with considerably much more freedom to focus on one thought at a time, I started considering which theories to use. Among the interesting ones were *internal* and *corporate communication theory*, *employee engagement*, *human resources*, *social*, *cultural* and *industrial/organizational psychology*, *escalation of commitment*, *single/double loop learning*, *decision making*, *cross cultural management* (especially Hall's *high and low context cultures* and Hofstede's *cultural dimensions theory*) and a few others.[4] But sitting and attempting to compare these theories, I realized that all of them would work about equally well to explain certain aspects of my experiences. On the other hand, that also meant that all of them were equally bad at explaining the full story. Knowing that the company had spent considerable resources on finding answers to how to increase efficiency, I came to the conclusion that while my short thesis might accurately compare a case study with academic theory, it would do very little to find the causes for the underlying problems in this particular organization. Instead, I started to view my experiences not as a sequence of separate events, but as one continuous story which contains a small set of consistent "truths."

After I had found a few theories that seemed to explain what was going on at a more profound level, both my academic supervisor and I gradually began to understand that the strength in my thesis is my unique position to see every part of the daily activities in the company, something researchers normally can't as they are not part of the organization in the same way. Trying to "cover everything" didn't make sense to us from a traditional academic perspective, but when writing an autoethnography it suddenly made perfect sense. My story isn't one about communications problems or organizational learning. It's about what I saw, heard, and felt! It's about my relationship to my colleagues, the problems we faced and how we (for better or for worse) decided to deal with them!

During the process of writing, I spent a lot of time going through events in my mind and trying to see how different theories would match. I also got help focusing my thoughts from some of my former colleagues who supported me, and of course from my academic supervisor. In a way similar to when I was keeping my diary, I needed to write to find out what sounded reasonable and what I believed in.

A difficult part of the work was to decide what to tell and what to leave out. I had to choose to tell a representative part of the whole story, while leaving out parts as they could sound incriminating or simply didn't fit in any of the sections.

2.5 Ethical considerations

The thesis primarily focuses on the company. Since the topic was chosen by me without explicit permission from the company, I will present the company in an anonymized format to avoid drawing unnecessary attention to it. Furthermore, I'm bound by my non-disclosure agreement and may

[4] It's out of the scope of this thesis to go into any of these theories in detail, or even give advice on literature for further reading. A search on the Internet should be enough to give the reader a quick introduction to any of these concepts, however.

not provide information that may harm the company. Hence, I will not reveal any details of the way they work, and any problems such as meeting deadlines are already known by shareholders. The thesis should instead be seen as a proof of the room for improvement.

In order to successfully analyze an organization, I need to have a close look at the people within it, but that is where the research ethics gets tricky. Tolich (2010) presents a very critical view of the ethical aspects of autoethnography. He criticizes the way that for example Ellis chooses to describe others. He argues that although what she's trying to teach her students is generally good advice, she doesn't meet her own standards, when she for example chooses to skip parts of an autoethnography when reading it to her mother.

Tolich (2010) further argues that retrospective consent puts an unnatural pressure on the participant and instead advocates process consent, where the participants are continuously asked if they still want to participate. He proposes ten guidelines touching upon consent from participants, consultation from others and vulnerability of everyone involved. It should be mentioned that the Congress of Qualitative Inquiry (2007) lists standards of conduct when doing research on human subjects.

My subjective experience is, however, that the articles that emphasize consent the most are published in medical journals. That's the case with Tolich, for example, who was published in *Qualitative Health Research*. It's not obvious that corporate research needs to meet the same standards for concern for the participants that medical research should.

Spicker (2007) argues that consent—meant to protect people's privacy and rights—can sometimes be unnecessary. This might be the case in companies, as the participants are acting officially on behalf of their roles in the company. Besides the case about research on public subjects, Spicker also says that consent might be unnecessary when rights have been violated by the participant, as there are other ethical concerns at play than simply research ethics.

For this thesis, some of my colleagues knew about the project and supported it, but I never asked some of the more prominent participants for consent. My idea is that this is a study of the company, its culture and and teamwork. To accomplish that, I need to tell the story using colleagues as participants, but these colleagues act in their official roles. If, at any point, these participants step out of their official roles and act in a way not sanctioned by the company, then we instead deal with a case of abused rights. Either way, the story can be told without informed consent according to Spicker.

As Spicker (2007, p. 3) points out, research often has a public function.

> Criticisms of the actions of people in authority do not require their consent, and indeed the integrity of research could be jeopardised by the act of seeking consent. Observation, recording, and criticism are not only sanctioned; if there is an ethical bar, it is that it is illegitimate to put stumbling blocks in their way. (Spicker 2007, p. 3)

Although the quote primarily refers to government officials, the author also seems to promote such research within for example commercial organizations.

On a note of transparency, I want to point out that I hope that this thesis will not be perceived as hostile towards individuals. In order to allow greater transparency, I chose to write in English to avoid using the language barrier as a way to conceal my findings. I've also been assisted by friends from the company, with whom I've spoken both before and during the process of writing. Two of my former colleagues have been reading an early draft of this thesis to help me find potential weaknesses and mistakes.

3 Theoretical framework

Three levels of theories will be used in the analysis, on an organizational level, a group level and an individual level. These are interconnected, where the problems on the organizational level allow for problems on the other levels too. As I will attempt to show, the organizational problems

are the most important in explaining the situation, but some people are made better off because of how groups and individuals are allowed to act, which keeps the organization in a Pareto optimal[5] state.

The theories presented here were picked individually, although they have some links to each other. The organizational theory was revolutionizing when it was first published in that it explained that organizations might create solutions before knowing the problem, something which my company obviously did. The group theory helped explaining another well-known tendency at the company: that one person at a time was blamed for all the problems in the organization. The individual level theory, finally, offered a plausible explanation for the polarized opinions about some individuals who were either admired or strongly disliked, depending on whom you'd ask.

3.1 Organizational behavior

Cohen et al. (1972) introduced the concept *organized anarchy* and the *Garbage can model*, which are typically used together. Organized anarchies are characterized by (1) *problematic preferences*, (2) *unclear technology* and (3) *fluid participation*.

Problematic preferences means that the organization lacks an action plan. The guiding principles are a collection of ideas, and the organization discovers its preferences through action. (Cohen et al. 1972, p. 1) One example is universities, which have goals such as research and education. These goals seem very abstract when compared to most companies, which are providing very specific goods and services. (Hayes and McGee 1998, p. 30)

Unclear technology means that the members of the organization lack an understanding of the tools and processes in use. (Cohen et al. 1972, p. 1) Technology is in this context not limited to physical tools. In education, for example, faculty members rarely agree on the best method for conducting education. (Hayes and McGee 1998, p. 30)

Lastly, the organization has fluid participation if people are moving between the teams and in and out of the company, creating unclear boundaries. (Cohen et al. 1972, p. 1) A decision depends on the people involved, and a committee that brings up the same question over and over might very well change its decision. Even if the committee members remain the same, its customers and stakeholders are likely to change over time. (Hayes and McGee 1998, p. 30)

Traditional theories assume that management describes well-defined goals and that the organization works with well-known technologies. Choices are assumed to be made by a linear chain of events starting with a search for decision alternatives, continuing with evaluation and ending with a decision. Since the theory of organized anarchies rejects those assumptions, it needs to explain how an organization could still make progress without the support of that kind of structure. (Cohen et al. 1972, pp. 1–2) This is how the authors summarize the complicated situation:

> [An organized anarchy] is a collection of choices looking for problems, issues and feelings looking for decision situations in which they might be aired, solutions looking for issues to which they might be the answer, and decision makers looking for work. (Cohen et al. 1972, p. 2)

Cohen et al. identified four relatively independent "streams":

1. *Problems:* Needs that have emerged inside or outside the organization. They could concern just about any field regarding human nature. Problems require attention.

2. *Solutions:* End results of someone's work. Solutions often exist before anyone has thought of combining them with a problem. For example, GPS was invented for military purposes, but is now a solution to the problem of finding your way with your car without having to read a map. As the authors put it, "you often do not know what the question is in organizational problem solving until you know the answer."

[5] A state is *Pareto optimal* if it can't be changed without making at least one individual worse off.

3. *Participants:* The structure of the organization changes continuously as participants come and go and the energy they can spend varies. Participants sometimes have preferences for certain problems or solutions.

4. *Choice opportunities:* The organization is expected to make certain choices at different times. These choices can be about how and when to spend money, whom to recruit or promote and so on. (Cohen et al. 1972, p. 3)

In the organized anarchy, these streams cause redundancy since problems and solutions often appear at different times and disconnected from each other. When no match has been found, the ideas and whatever has been associated with them are thrown into the garbage can. But the garbage can is never emptied and nothing is thus discarded forever, and the organization frequently returns to the garbage can to dig around for old solutions that can now be matched with problems, possibly as a result of the new set of people on the team. (Cohen et al. 1972)

Cohen et al. (1972) as well as Hayes and McGee (1998) use universities in their examples of organized anarchies. However, Cohen et al. (1972, p. 16) point out that these behaviors can at times be observed in almost any organization. It just happens to be very common within universities. Furthermore, they explain that the Garbage can decision model shouldn't necessarily be perceived as a symptom of a defective organization, since it merely tends to kick in when other prerequisites for decision making are not present. One of its strengths is its ability to explain why even organizations with conflicts and goal ambiguity can make progress. Nevertheless, it's clear that the Garbage can process does a poor job resolving problems.

To further explain the merits of organized anarchies and the Garbage can model, DiBella (1992, p. 56) states that there are problems with assuming that organizations work towards a common goal. Some companies are required to adapt to changes and can't properly plan ahead. Other organizations might for one reason or another be difficult to control and even if management tries to convey their ambitions, these messages might intentionally or unintentionally be rephrased along the way.

DiBella (1992) studied a non-governmental organization which can be categorized as an organized anarchy, and even though the article describes an organizational change rather than just the present state of an organization, his findings are relevant for understanding the concept even without organizational change. One of the most important points he's making is that managers who don't understand the nature of the organization they're operating in might be deceived by the apparent consensus since, in the absence of a common goal, it promises nothing about how people will interpret new information on their own:

> Managers who pursue a modernist tactic of developing shared vision can get fooled or misled by early agreement or lack of disagreement. Initial agreement can give a false sense of security about the potential for problems later on. (DiBella 1992, p. 64)

Hayes and McGee argue in a similar way:

> If we don't understand the politics, the motivations of colleagues and customers, the ways decisions are made or unmade, and the many effective management techniques available to us, then no amount of money or technology will ensure full success for our systems. (Hayes and McGee 1998, p. 29)

Cohen and March (1974, pp. 207–215) provide a set of rules for leadership in an organized anarchy, in order to achieve something of value. In short, they recommend (1) spending enough time to build a proper case and in the meantime earning recognition for the effort, (2) persisting and not taking no for an answer since it might be different the next time, (3) being prepared to give status and credit to others in exchange for having your suggestion implemented, (4) bringing outside people into the decision-making process to stir things up and reduce the effects of office politics and bureaucracy, (5) producing many proposals—instead of focusing on a single one—while accepting that some will be rejected, (6) providing garbage cans to collect garbage early on

during a meeting, while saving the serious concerns for later, as the likelihood for implementation of a proposal decreases over time as the amount of connected garbage increases, (7) avoiding confrontation in order to maintain a benevolent atmosphere and (8) keeping records of historical events in order to let the company learn for the future.

3.2 Group behavior

A *scapegoat* is a social role, which assumes guilt from someone else. Historically, a magical ritual was performed to transfer the sins of the tribe onto a goat, which was then killed or driven off into the wilderness. In this way, the tribe was purged of its guilt. (Gemmill 1989, p. 406) Wilson (1993) builds on previous work where he identified the act of *corporate scapegoating*, a defense mechanism to avoid blame by aiming it elsewhere, usually a particular individual. He argues against *methodological individualism*, which states that the *intentional agent* must be some one individual, or otherwise no one in particular is responsible. Wilson's opinion is that a corporation can act intentionally on its own, and he expresses this with the phrase *corporate agent*.

As a comparison, Wilson (1993, pp. 779–780) uses the *systems approach* to family therapy, where the therapist treats the entire family as one patient. No individual family member is seen as the cause of problems with the family's well-being. However, it's common that a family will have identified one of its members as the sole dysfunctional individual. This is known as the *identified patient syndrome* or scapegoating. A therapist must be careful not to accept such explanations.

Although Wilson primarily uses a fictional character to exemplify, he also brings up the case of the oil tanker Exxon Valdez that spilled millions of gallons of crude oil in the sea off the shore of Alaska. The captain was allowed to command the company's largest tanker even though he had a reputation for being intoxicated, but after the accident his employment was terminated and he was indicted for negligence. Wilson notes that the company avoided accusations of gross negligence, and adds that the severity of the accident was due to the fact that the Exxon Valdez was a single hulled tanker. A double hulled tanker would have been capable of reducing the damage or even preventing it completely, but the construction of the ship was clearly nothing the captain could have affected. (Wilson 1993, p. 784)

Gemmill (1989) describes the phenomenon of scapegoating from a clinical perspective. He describes the concept of identified patient more in detail, and emphasizes that it's an unconscious act of distributing covert roles in a group. Scapegoats are often used by authoritarian leaders in a group, and the member assigned the scapegoat role is often an unconsciously willing victim. As the role is undiscussed in the group, a *self-sealing nonlearning* of the dynamics occur, and the system remains unchanged. The group is furthermore unwilling to let go of this equilibrium, as that would stir up emotions. Moreover, if the scapegoat would leave the group for some reason, the group tends to quickly reassign the role to someone else. The scapegoat usually turns more passive and silent, something which is accepted by the rest of the group as it also silences emotions within the group. Gemmill also speaks about more complex processes, where several scapegoats take turns as time passes and the group is working to solve its issues. In the end of the article a proposition is offered, that awareness of the underlying conflicts will help the group work through its covert issues, and make the group more constructive and productive. While this is an intriguing proposition, the author points out that the area needs more research.

The theories put forward by Wilson (1993) and Gemmill (1989) complement each other well, but it should be noted that Wilson doesn't explain whether a corporate agent is unconscious of its actions in the same way as the clinical groups described by Gemmill (1989), or whether the possibly different environment within a company makes individuals deliberately and consciously advocate the use of scapegoating to meet company requirements.

Runge (2009) points out that scapegoating often occurs as a result of a failed IT project. She

mentions time as a facilitating factor, as people will leave the company and others will forget who was accountable for the investments and the decisions. With everyone responsible and no one accountable, scapegoating is the organizational solution. Runge argues, however, that the causes generally lie with the C-level and senior executives and presidents, who should make sure that the project is sufficiently analyzed, scoped and communicated. She makes the connection to communication theory clear by stating that "Information cannot be expected to be communicated via osmosis or hearsay."

Runge continues by saying that those who were responsible for identifying and collecting requirements need to be empowered and accountable. They should have been selected because of skills, and not because of seniority or self-appointment. She concludes that it's important that extensive requirements are collected and correctly documented, but that both parties often simply assume that this has been done. This gives rise to situations where scapegoats are sought out.

On an individual level, it's possible to imagine how accusations implying that someone bears the sole responsibility for a project failure could have severe ramifications for the career, but a remaining question is what kind of implications this activity has for the company. Ordoñez (2009) has developed an economic model for scapegoating, the mathematical details of which are, albeit complicated, not entirely necessary for the understanding of the concepts. The model introduces the ideas of *nested activities* and *nested reputation*. These nested activities are meant to substitute or complement an original activity that generates reputation. Although the nested activities may be irrelevant from a customer point of view, they are able to affect reputation within or outside the organization.

According to the model, reputation will be affected positively by success and negatively by failure. However, after failure, a nested activity such as scapegoating can decrease reputation loss. Hence, a superior will attempt this nested activity after every failure, to maximize the final level of reputation. The problem with efficiency arises when superiors suspect that the risk of failure increases. In this situation, blaming becomes a more secure and hence more attractive way to build reputation, and since experts are harder to blame for failures than non-experts superiors become less willing to hire experts during bad times, as the expected reputation loss after a failure will be greater. (Ordoñez 2009)

Safire (2007) wrote an article, contrasting a scapegoat with a *fall guy* (with a unisex implication of "guy") or someone "taking the fall," which primarily discusses the origin of the expression. The article mentions the definition that a scapegoat is entirely innocent, while a fall guy is somewhat complicit but still assumes a larger portion of the blame than what's fair. Towards the end of the article, different senses of the expression are touched upon, ranging from a fall guy being an innocent victim to someone who willingly or, because of fear, knowingly accepts the blame. An understanding of the complexity of the definitions is helpful when applying the theories. However, the article also makes it clear that no unified interpretation of a fall guy exists, and I choose not to distinguish it from a scapegoat.

3.3 Individual behavior

Lubit (2002, p. 128) explains the distinction between *healthy narcissism* and *destructive narcissism*, which are different both in their causes and their symptoms. A healthy narcissist has a secure self-esteem and empathizes with and inspires confidence in others. A destructive narcissist lacks a stable self-esteem, but will appear confident. The lack of self-esteem causes the narcissist to devalue and envy others, and when put under pressure the productivity decreases. A healthy narcissist often enjoys admiration, power and wealth, but a destructive one is obsessed with them. Specifically, the defining properties of a destructive narcissist are (1) grandiosity[6], (2) an overblown sense of entitlement and (3) lack of concern for others.

Kets de Vries and Miller (1985, pp. 588–590) point out that it's not a matter of whether nar-

[6] defined by Lubit (2002, p. 128) as "inflated sense of self-importance, arrogance, preoccupation with power and wealth, excessive seeking of admiration"

cissism is present or not, since everyone exhibits some level of the characteristics and that they are necessary in order to function efficiently. Hence, we should instead focus on the degree of intensity and the position on the spectrum between healthy narcissism and pathology to find the extreme cases.

Destructive narcissists value others by how they can contribute to their personal needs, and their sense of entitlement allows them to exploit others and borrow things without asking. They see nothing wrong with this and are unaware of how their behavior affects the company as a whole. Their envy makes them devalue others, but they often do it by overstating actual personality flaws which everyone has. They keep a few chosen people close, requiring total devotion from them. These loyal friends will help to praise the narcissist's competence even more, and the apparent self-confidence allows them to move up the ranks, because people tend to believe that self-confidence is the result of real competence. (Lubit 2002, pp. 128–129)

The subordinates of a destructive narcissistic leader are usually constrained by a lack of clear expectations, instructions, recognition, concern and encouragement from their superior. Submissive subordinates can get promoted, but the talented ones are likely to leave. Ideas from subordinates are often ridiculed, and interests rapidly change. The destructive narcissistic leader typically doesn't want to focus on details, and might lack what it takes to follow through, making implementation difficult. (Lubit 2002, p. 130)

As a cause for destructive narcissism, Lubit (2002, pp. 132–133) presents two explanations: *psychodynamic theories* and *social learning theory*. Psychodynamic theories argue that events in the early childhood caused a fragile self-esteem and gave rise to a defense mechanism with the need to devalue others and behave grandiosely. Social learning theory, in contrast, argues that much of our behavior is learned from observing others, and reinforced through positive feedback, but possibly held back by standards and rules about how we are allowed to behave and where the limits are, learned during early childhood. Lubit says that both of these theories have their merits, and distinguishes between *learned narcissism*, caused by events according to the social learning theory, and *psychodynamically based narcissism*, in turn caused by events in the childhood described by psychodynamic theories. Of these two, learned narcissism is easier to deal with, since it's a learned behavior that can be changed as the leader understands that it's unacceptable.

Kets de Vries and Miller (1985, pp. 590–593) offer a comparable explanation with three categories. *Reactive narcissism* is caused by a problematic narcissistic phase in early childhood, *self-deceptive narcissism* is caused by overstimulation during childhood because of unconditional love from the parents regardless of the child's actions, and healthy or *constructive narcissism* can occur despite a normal and stable childhood. The first two types would thus correspond to Lubit's psychodynamically based narcissism and the latter to his learned narcissism.[7]

Destructive narcissistic leaders can rise in organizations if the organizational structure and culture permit it. Since the narcissistic behavior is usually directed towards subordinates, the behavior is often not noticeable until after promotion, and even then it's difficult for the superiors to find out about it. Important factors that make an organization vulnerable to destructive narcissistic leaders are (1) hiring practices that rely on recommendations and performance during interviews, (2) a culture that tolerates the behavior, (3) a performance measurement system that doesn't track long-term development and sustainability, (4) the presence of leaders exhibiting narcissism already and (5) processes that don't require the leader to work in teams. (Lubit 2002, pp. 134–135)

According to Lubit (2002, p. 135), destructive narcissistic leaders make more harm the higher they are in the corporate structure, but can do serious harm also at lower levels. They pursuit self-interest, and can selectively neglect tasks and people. The most capable people pose a threat

[7]The different categories have very specific personality traits, and Kets de Vries and Miller describe them in detail, but it's out of the scope of this thesis to do a psychological study.

to the leader, and are thus more likely to be neglected. This makes talented and skilled people more likely to leave the company or move to another unit.

Lubit (2002, p. 136) moves on to describing warning signs to look for before making a decision about promotion, notably including scapegoating, and then gives advice for how to recognize destructive narcissistic leaders and how to cope with them. The best way to identify problematic leaders is to introduce *360-degree feedback*[8] and arrange so that feedback from subordinates can only reach their superiors in an anonymized form and/or if a majority has the same kind of criticism.

If a destructive narcissistic leader is valuable enough to the organization, then Lubit (2002, p. 136) argues that it might be a small price to pay. Oxford (2012a,b) doesn't agree. While he doesn't speak about narcissistic leaders in particular, he explains how someone who was once an asset to the company, a *brilliant talent*, can become a liability as the company grows, a *brilliant jerk*. His advice is to get rid of this person as soon as possible, but estimates that it usually takes around 1.5 years of wasted time for this to happen.

> I can tell you from personal experience that coddling the Brilliant Jerk – letting him work from home, consoling him, giving him special assignments – does not work. It just kicks the can down the road. At my company, I was worried about the impact his firing would have on other employees who had shown him respect. To my surprise, the reaction was, "What took you so long?" (Oxford 2012a)

> The Brilliant Jerk sucks the life out of a company. Every minute you spend consoling the Brilliant Jerk is a minute you take from customers and, by the way, it is the customers who pay the bills. (Oxford 2012b)

If the company chooses to take action, then a confrontation is usually necessary. This confrontation should not be initiated by subordinates, but by superiors or peers. When dealing with severe psychodynamically based narcissism, direct confrontation can cause matters to get even worse, though, and outside professional help might be necessary. Lubit proceeds with advice to subordinates to destructive narcissistic leaders on how to deal with the stress and frustration. A subordinate shouldn't try to explain to their superior how the situation affects them, and should avoid arguing with them, gossiping with them, borrowing from them or lending to them. Rather, the subordinate should appear to admire the superior. (Lubit 2002, pp. 136–137)

> Try to obtain written directions whenever possible, since they decrease the room for uncertainty and complaint about you. Document your work so you can defend yourself if they criticize you for failing to do your job properly. Document interactions and the course of events so that if you need to defend yourself to someone higher up, you have the means to do so. (Lubit 2002, p. 137)

Lubit then explains how to change the situation by first leaving the unit:

> Once you are out of their unit, report to superiors how they treat people. If possible, do this in collaboration with others who can validate your statements. Informing superiors of the problem will help the company as a whole and improve the working environment for all. (Lubit 2002, p. 137)

Kets de Vries and Miller (1985, pp. 599–600) agree about introducing better processes for performance reviews and recruitment, but they say that it's very difficult to change the behavior of a narcissistic leader, and that it's better to attempt reassignment or reduction of influence. They also specifically warn about allowing insecure and inexperienced managers to work closely with a narcissistic leader. A strong and confident manager might, on the other hand, act as a counterweight to mitigate some of the negative effects.

3.4 Summary of theories

The three levels of theories I've been presenting are the organizational level, the group level and the individual level.

[8] obtaining feedback from not only the subordinates but also through self-evaluation and from peers, supervisors and sometimes even customers and suppliers

On the organizational level, I've picked theories of organized anarchy and the Garbage can model. These explain how organizations without coordinated efforts can make progress. They do it by coming up with solutions to questions that no one has asked yet, and then storing them until a time when it's possible to make a match. These organizations are bad in terms of productivity speed, however.

On the group level, I've chosen scapegoating, which is what happens when a group can't deal with the fact that they have a shared problem. Instead, the group finds a scapegoat, that will assume all the guilt from the group. Organizations that do this might choose to recruit people who will be easier to use as scapegoats.

On the individual level, I'm describing constructive and destructive narcissism. The root causes might be different depending on the type, but narcissism can have several negative consequences including lowering productivity and driving off talented employees.

4 The story of a middle manager

My experiences from the company are described in subsections based on different themes. Hence, the subsections are in most cases chronological internally, but the different subsections don't follow each other in a chronological order. The section starts with a description of some of the people mentioned in the story.

In some cases, I mention dates or months to help the reader keep track. To put these into perspective: I worked 16 months for the company, between February 1st 2012 and June 7th 2013.

4.1 The main players in this story

A simplified organizational chart from the time I left the company can be found in Figure 1, where the internal structures of the Business, Growth and Finance departments have been left out. When I first joined the company, the structures weren't as well defined, and all the engineers belonged to the Product department. The Tech department was formed around November 2012, but information about the switch was unclear and even at the time I left, there were still people who used the name Product department when speaking about the developers. Quality assurance was removed in February 2013. The structure where the CTO would report to the CPO was something that I heard from the CPO that the CEO wanted. The CPO, on the other hand, preferred to keep the direct responsibility over all the developers, which he had at that time. Hence, he saw no need for a CTO whatsoever.

Not all of the people described below chose to use these exact titles. For example, there were people who called themselves Director although they were really at the vice president level (C-level). I choose to use these commonly accepted titles to avoid confusion. Also, all of the given names are pseudonyms.

The CEO Born in China, but moved to Sweden with his parents as a 7-year-old. He started to study at KTH to become an engineer, but dropped out of university to move to China and founded this company with two others. When I worked at the company, he was in his early thirties.

The CPO The Chief Product Officer was a French citizen in his mid-twenties. He was in charge of the Product department which, at the time I started at the company, included both the developers, the Operations engineers and Quality assurance engineers, plus a small administrative staff. Since he didn't have any designers to help him, this made the CPO (1) product owner, (2) the only interaction designer, (3) the only graphic designer and (4) the head of all the engineers.

Later, partly because of pressure from me, the developers and Operations were transferred from the Product department and formed the new Tech department. A few months after this, the company terminated the employment for the Quality assurance engineers, who were at the time still in the Product department, and distributed the tasks between the developers and Operations. At the time of my resignation, the Product department consisted of only the CPO, his assistant and a newly instated technical writer.

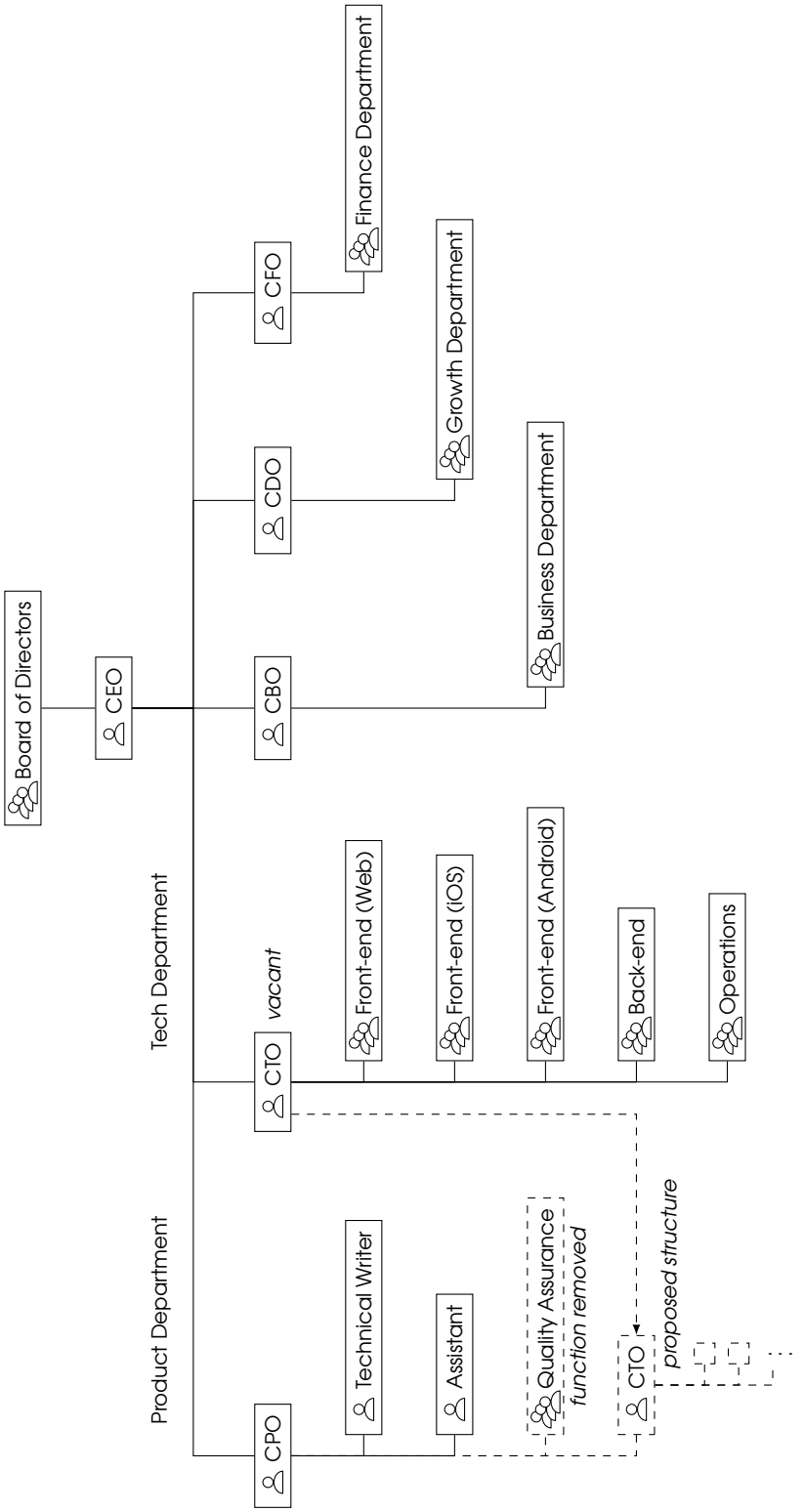

Figure 1: A simplified organizational chart

The CDO The Chief Data Officer was a Swede in his early thirties. He had a background in environmental organizations, and started to work for the company as a photographer. As a CDO, he was in charge of the Growth department, which focused on strategies for developing the user base. Because of the lack of a human resources function within the company, he had also been assigned the responsibility of HR manager (probably since it could be seen as just another kind of company growth). I think it's impressive that he could get promoted from being a photographer to the head of a department, but I don't know any details about this promotion.

Thomas/The interim CTO The position of Chief Technical Officer had been vacant since before I joined the company. Thomas, a Swede in his mid-thirties, was the friend of mine who had recommended me to the CEO. He had held an influential position in a popular Swedish start-up in the online music industry, but was now working as a guest professor at a university in Texas while he acted as an advisor to the board of directors at the company. His intention was that I would become the new CTO. I started at the bottom as a developer, and moved up and took charge of the back-end team but since I hadn't earned the trust that I needed from some of the others I was never allowed to become CTO. Instead, Thomas had to fill the position as interim CTO while the company was searching for someone else. This meant that he flew from Texas to Beijing about once a month. I'll refer to him as either Thomas or the interim CTO depending on the context.

The Chairman The chairman of the board, a Swede in his early sixties with a business degree from the Stockholm School of Economics, had much experience of the Asian markets and used to be the head of the Swedish Trade Council in Taiwan. Still living in Taiwan, he was flying back and forth to Beijing, and he was heavily involved especially when the company was searching for new investment. He had high integrity, a business mindset and valuable experience, but unfortunately lacked a deep technical understanding of software development. Towards the end of my employment, he took a considerably much more active part in the day-to-day management of the company, likely because of a series of setbacks that had made board members and shareholders partly lose faith in the company.

Randal Randal was a British citizen, who moved to China because his Canadian girlfriend had problems attaining a UK work permit. In the UK, he had been working as an independent contractor, and had a lot of experience from all kinds of companies. He became one of my closest friends in Beijing, and his experience was a huge asset to me in my work. Unfortunately, the company failed to recognize the potential that I saw in him, and he was marginalized to the point where he finally did the only reasonable thing and resigned. Towards the end of his employment, he questioned the work he was doing and said that he felt like he was "rearranging the chairs on the Titanic." The story of Randal is an important one in understanding how the company operated, and it will be told in subsection 4.11.

Frank When I joined the company, Frank was a *Scrum Master*, that is the manager of the Scrum team (see page 3). As the processes changed and the teams weren't using Scrum anymore, Frank was reassigned and moved between responsibilities. For a while, he was assigned to helping me with planning, but I felt that he added a lot of work which consisted of different ways for me to report my progress to him, without actually providing any assistance with the tasks that were already delegated to me. This led to a conflict, since I couldn't keep working like that. He also took the role of gatekeeper to allow the developers to focus on their tasks without being interrupted by external requests, and towards the end of his employment he was in charge of Quality assurance. Because of the vague and changing responsibilities, it was difficult for people to know what he was really doing. This was probably one of the reasons why his employment was terminated in February 2013.

Dan Dan was the only Chinese developer who worked for the company during my entire stay in China. Having started to work for the company in February 2011, he was also the most experienced developer on the staff. Dan worked in the back-end team, which I was leading.

"The French group" What I and others called the French group wasn't a formal group at the office, but rather a group of colleagues and friends. They had recommended or recruited each other, and spoke highly of each other in front of others. My understanding is that many of them were from the same school in France. The French group consisted of mostly, but not exclusively, people in managerial positions. They were the CPO, Frank, the two iOS developers, the lead web developer and during the fall of 2012 also one back-end developer, who worked with us for a few months. A group of predominantly Swedes were, in a similar way, referred to as "the Swedish group." This was the group I was labeled a member of, and the French explanation for the internal problems of the company seemed to be that the Swedish group consisted of *thinkers* and the French group of *doers*. Still, it's important to remember that the division into groups by nationality wasn't strict, and there were both Swedes and Frenchmen who weren't regarded as members of these groups.

4.2 How I ended up working for the company

This subsection will shortly explain my background. It's relevant both because it gives a context to my analyses and my way of thinking, and because it's something my colleagues who are mentioned in this story knew about me. A lot of it had been announced to the staff before my first working day.

Since I was young, I've been interested in computers, but when I applied to KTH I had already decided that I wanted a career with a focus different from programming. Thus, I started to study engineering physics. But my studies began before I had done my mandatory military service, and having returned from the army I felt bored by the physics courses, and in 2005 I applied to SSE. Being a decently good student, I got the chance to go abroad twice for exchange semesters. Still, with little left on my business curriculum except for my thesis, I realized that something was missing and that I really wanted that engineering diploma as well. In 2010, I decided to switch from engineering physics to computer science, and was admitted into the second year because of the courses that I could transfer from my old program.

The switch was one of the best decisions I've made. While the business courses were overall fun and interesting, the computer science courses were truly enjoyable. I also managed to complete the second year during that fall, and made sure to be registered for the third year in the spring of 2011 so that I could write my bachelor thesis. This led to my being admitted to the master's program in computer science in the fall of 2011, and with some effort I finished basically everything apart from the master thesis before the spring of 2012.

It was during that fall that Thomas started to ask what I wanted to do as my first job. Although I had spent most of my time from 2005 at SSE, there was still something I continued doing at KTH: In the fall of 2004, I had joined the Nordic Collegiate Programming Contest with a one man team (you were supposed to be three). It just seemed like a fun thing at the time—you'd spend a Saturday at KTH solving programming puzzles, getting free pizza and a t-shirt. I think it surprised me more than anyone else that I did well enough to be teamed up with two other people and sent to the North Western Regional Programming Contest[9]. As I continued to compete over the years, Thomas was my coach. Since I needed some advanced courses to improve my problem solving skills, he had also been my teacher. But after finally having reached the World Finals in 2009, I no longer qualified to be a contestant in a student competition because of my age, the duration of my studies as well as the number of times I had competed in NWERC. From this time, whenever

[9]NWERC was the regional finals for students from Belgium, Luxembourg, Great Britain, Ireland, Iceland, Norway, Sweden, Finland, the Netherlands and (at that time) parts of Germany.

I was in contact with Thomas, he was more like my mentor. In the fall of 2011, the reason why he asked what I wanted to do was that he knew of a company in need of recruiting.

The company Thomas had in mind was run by a former coursemate of his. They had recently lost their CTO and needed to find a new one. The problem was that the company was situated in Beijing, China, and my focus had since several years been in Eastern Europe, specifically Russia. One of my exchange semesters had been to St. Petersburg, and I had spent more than three months in Moscow to pass a certification exam in Business Russian. I hesitated, but then allowed him to give my contact information to the CEO of the Chinese company.

The CEO called me several times. I didn't get the impression that it was as much to get to know me and see if I was good enough, as to just convince me to join the company in China. I guess the recommendation from Thomas was sufficient for the CEO, and he didn't feel like he needed to hold a proper interview. But given that China wouldn't have been my country of choice, I had a number of concerns. Among these was my desire to finish my theses in computer science and business. The CEO assured me that that wasn't a problem, if I could just join the company as soon as possible. In the end, I found myself having accepted his offer, but with the condition that I would get to take my exams in January. Hence, February 1st 2012 was my first working day at the company.

4.3 An outline of my time at the company

As a new employee, I started reading the internal documents I could find, and was then assigned the task of cleaning up old back-end source code. I quickly found out that the back-end code was a mess and that although some improvements could be made, it would do very little to simplify future development. It also taught me a valuable lesson about myself: If I realize that something is so poorly designed that it would save me time if I just started over from scratch, then I can't find the motivation to work with what's already there. Instead, I started looking for weaknesses and designed two simple but severe hacker attacks that no one had thought of before, and then demonstrated them to the Product department. I believe that one of the reasons for why this could occur was the desire to avoid standardized solutions and instead coming up with our own custom tailored solutions, or in other words "reinventing the wheel." This is an attitude that I came across several times, and I spent effort fighting it, since standardized solution and already existing programming libraries have usually been tested thoroughly, and chances are that you'll avoid the most common pitfalls by taking advantage of them.

After a while, I started leading my own Scrum team as the company decided to split the one team that they had into two, but I was frequently asked to adjust the sprint length, to make adjustments to the stories that we had already accepted into the sprint or to reprioritize between them. Since Scrum as a framework doesn't allow this behavior, as it impedes development and reflects badly on the team, I chose to switch to Kanban, which doesn't measure the time span during which work must be completed and allows tasks that haven't been initiated to be reprioritized at any time, that framework worked better in our volatile situation. However, this caused the Product department to work with several different frameworks for software development, something that was probably part of the reason for the isolation of the back-end team, which we started to notice.

The deal was that I would work part-time at first in order to finish my business thesis. I was going to introduce similar agile methods used in software development in the Growth and Business departments. This project was initiated, but the heavy workload (described below) made it difficult for me to follow up, and the departments I was planning on studying didn't use the methods in the intended way without my guidance. Since the company was under pressure to deliver, I was asked to work hard for a while, and was told that I could focus on the thesis later. Furthermore, the situation was often described as being bad enough that the company was facing a threat of shutdown either by shareholders or be-

cause of a lack of funds already within 2–3 weeks, and the only way to save it was for everyone to work overtime. Similar reasons were stated several times, and I was also dependent on the company to keep going if I wanted to finish the thesis. Hence, I typically worked seven days a week, and stayed late at night. In the end, there wasn't much of a thesis project to return to. Sadly, the effort I put in wasn't reflected in my paycheck, as the company was paying 50 percent of my negotiated salary until the end of the summer, according to the deal about part-time work. The money I lost during this time, corresponding to 3.5 months of full-time salary, has not been compensated by the company at the time of writing.

I'm not sure if there was ever a formal decision, but I ended up being recognized as the head of back-end somewhere around the beginning of the summer. I had to work with the French group that otherwise dominated the leadership roles in the Product department, but since they had stopped asking me to join them for lunch or coffee my chances to communicate with them were limited to meetings and e-mail, both of which were tools that they were reluctant to use.

As the head of the back-end team, I got the responsibility of designing and implementing a new API. The company had earlier had the policy of using simple tools that a lot of people knew for building its systems. But while the tools were simple to learn, they were hardly the best tools to do the job. Thus, I made plans to upgrade the back-end system to state-of-the-art technology. My opinion is that when using the proper tools, the work will be faster and the result better even though the learning curve might be steeper. Also, if someone we would otherwise have considered recruiting was unwilling or unable to learn a new programming language, then we wouldn't have wanted that person anyway. But in the end, the technology switch was put on hold for several months, to accommodate short-term goals.

Unfortunately, after taking charge of designing a new API, the summer and fall would turn out to be a never-ending story of miscommunication, delays and criticism. From my perspective, I wasn't asked "When can you have this feature done?" until company management had already promised the shareholders a certain delivery date. I felt forced to accept the same deadline that management had already decided upon, giving a promise of what I could deliver under "ideal circumstances," meaning that my entire team could work on the task without interruption. What happened afterwards was always that the team was assigned more work or was given a clarification which extended the scope of an existing task, that one of my team members was temporarily reassigned to another team, that someone simply got ill, or a combination of such conditions. Still, even with the decreased capacity we usually managed to finish the task without more than a two-day delay, and my team felt happy about our progress. For some reason, however, what we delivered was rarely accepted without modifications. I had done my best given the limited information available, but it was up to others to decide whether it was what they were hoping for or not. I will continue to discuss these events on page 41.

From the outside, it seemed like the back-end team could never deliver on time and that our work was substandard. The original idea, that I would become CTO, never materialized. The CEO told me that I had his support, but that the CPO had objections. The CPO, on the other hand, said that I had his support but not the support of the CEO. Furthermore, the CPO explained that even if the company would have a CTO, the CTO would have to report to him and thus not be on the management team, as this was what the CEO wanted according to him.

The communication problems between back-end and the rest of the Product department kept on for most of the fall, and despite the fact that my team members felt that we were doing well, it was hard to keep the motivation. The combination of the missed opportunity to write my thesis, the improbability of being promoted to CTO and the demand to apologize to people for my sloppy work, when I in reality worked about four times as much as I was paid to do, finally made me realize that I had no future with the company. In other words, I was at peace with my decision to leave the company. Before actually resigning,

however, I wanted to see the results of my effort, something that made me stick around for several more months.

During the spring of 2013, the lead mobile developer was fired. As far as I understand it, the situation was desperate and the shareholders needed to see a version of the iOS app with fewer bugs lest they lose their trust in the company. In the time that followed, there was a lot of talk about the developer who had been fired, and how bad he had been. While I could agree about some obvious mistakes that he had made, I felt that I didn't really know enough to share the opinion. As three other Frenchmen were fired on the same day, this didn't exactly improve the situation between the Swedes and the Frenchmen. The French group got the idea that racism was the underlying reason for the decisions to fire them.

As a first step towards solving the problems on iOS, I and another guy from back-end were temporarily moved to the iOS team to improve the app. Having to move made me lose control over back-end, and the team couldn't perform as well as before. The short-term solutions and the communication that never seemed to improve finally made me hand in my letter of resignation. My final working day was June 7th 2013, a little over sixteen months after starting at the company.

4.4 Building trust with the CEO

My relationship with the CEO was volatile. Of course, initially it was he who called me and convinced me to come and work for him, but I also found myself being questioned a lot, and we had more long, heated and—as it often seemed to me—meaningless discussions than I even want to remember. These discussions were rarely planned, and often summoned with the words "do you have five minutes?" Although I knew that few of the conversations would be any shorter than an hour and a half, I found it hard to respond to the actual question in the way I wanted. Yes, I did have five minutes, but not the amount of time that I knew it would take.

One cause for our long discussions might have been that we were both extremely stubborn. We were equally certain that we were right, and even though we'd have to spend a lot of time arguing, no one wanted to abandon his position. But to some extent, I have to admit that I appreciate that. We were both striving to find the root causes of our disagreement, and even though our paths weren't always the straightest, the tactics eventually led forward.

One day the CEO came to me and asked how a particular problem of scalability could be solved, in terms of how many servers would be necessary as the user base grows. I told him that what he asked for is physically impossible, but he maintained that our competitors did it. I started to explain how a server request is being processed and what the logical conclusions about scalability are. I told him that if the software is built well, then the server might be able to handle up to *four* simultaneous requests (a number off the top of my head, but it sounded reasonable at the time). It was a Saturday afternoon, and this day I had hoped to be able to do some of my own things that had been postponed for a while. Hence, after a long while of him insisting that our competitors had solved this problem I got annoyed and I remember thinking "I wonder if I can make him understand if I punch him," but with some self-control I managed to suggest "Fine, let's estimate it then." We searched online for available information on our competitors and other people's estimates about their number of users and servers, while making our own assumptions about usage and distribution of peaks. We used conservative guesses for server purchases and probably exaggerated numbers for user growth to make sure that the outcome wouldn't be too low. Our result? Approximately *four*, my initial guesstimate! The CEO sat quiet for a while, thinking. Finally, he accepted my answer that the problem had to be solved in another way, and I think of this as a turning point in his way of looking at me.

The worst conflict we ever had started when I was requesting an explanation of how a certain feature was supposed to be used by our users. Since there were no clear specifications of how this should work—merely an unspoken feeling of consensus among some of the people in charge—

he couldn't direct me to anything to read. Instead, he told me to start using three competing products, which he specified, and learn how they chose to implement the feature. I asked if it had to be these three or if others would do as well, but got an unclear answer. My point was that if we would copy the functionality in detail from a certain competitor, then it should be enough to only try that one out. In contrast, if I needed knowledge about several competitors to understand the concept, then it probably meant that the exact design wasn't clear enough to be understood just by looking at what the competitors did. I had already been criticized for not delivering what people were expecting me to, since I didn't have any specifications to follow, and I wasn't in the mood for another guilt trip. Besides, the CEO claimed that it would have taken me only five minutes to sign up and learn about the feature, but I believe it to be an understatement. If I need to use three different products, and mix the experiences from these in a way that my own company couldn't even explain in a document, then it would certainly take me much more than five minutes.

Although I had considered the matter closed of going to the competitors to find out what we should be doing, as I learned much later word of my "refusal to cooperate" reached Thomas. It also came back to bite me when the CEO late one evening sat down next to me at my desk to speak with me, and we accidentally touched upon the definition of that same feature, which I had still not made the technical designs for. That made him yell at me for over 20 minutes, and throw in some accusations about being lazy, sloppy, forgetful and unserious. I happened to have one of my subordinates sitting on the other side of the desk, and another guy across the room. They both left after a while, since they couldn't listen to it anymore, and the CFO even came over from another room and asked the CEO to have a word with him, but the CEO replied that he wasn't done yet. Under normal circumstances, I would have thought of it as kind of amusing that you could even imagine accusing someone of laziness when they're still sitting by their office computer at 11 p.m., but in this case I took it pretty hard.

The following morning, the CEO did in fact apologize and was hoping that things were alright. To be honest, they weren't okay, but I could accept his apology and hope that it wouldn't happen again, at least not in front of my team members. One more major incident would occur, however, already during the following week. I was at the office a Saturday evening, when the CEO came by at around 9:30 p.m. and asked me about a certain feature. I told him that it wasn't planned for the next release, and I was certain of it since I that very day had met the CPO in a doorway and asked him whether it was planned or not, something he denied. The CEO sat down next to me and started going through what we were doing. It turned out that there were multiple misunderstandings. The necessary changes were major, but he demanded that these would be solved by Monday, and asked me to call in my team to have them work on Sunday. This happened during a time when the Product department was struggling to work as a separate unit, without immediate influence by outsiders unless via the CPO or Frank, who was the appointed gatekeeper. Hence, I wrote an e-mail message to Frank at 10:15 p.m., a message which apparently opened Pandora's box. The CEO rushed into the room a few minutes later. He said that Frank was furious and that the CPO wouldn't even answer his phone. He had been forwarded the message with my request for confirmation and required me to send out a follow-up message with an apology for some of the words I had chosen, although they were little more than synonyms to the ones he wanted me to use. Then, he told me that he had also heard that I had sent another e-mail message earlier that day. The message he was referring to was one sent to Frank in the early afternoon where I had explained that I was buckling under the pressure and asked to be relieved of the responsibility as head of the back-end team, a request to which I never received any response. Unaware of its content, the CEO demanded that I show him the message, something I at first tried to dodge because of the sensitive content. After I had opened the message, we both sat there quietly. He was staring at my screen for a period of time that must have al-

lowed him to read the message several times over, and once he started speaking again his voice was softer. I had actually told him something similar just a few days earlier, but I think that he at the time attributed the statement to the heated discussion. Having read the message, he didn't mention the content. He just asked me to send the follow-up message, and then left. The response from the CPO was that the features that the CEO had requested should obviously be implemented. I had, of course, no way of verifying how obvious the features really were, since I had no written records of the decisions.

While I don't know it for sure, I believe that this event made the French group (to which both the CPO and Frank belonged) see me as an even worse problem, perhaps even a traitor. I could never figure out if it was because I had allowed myself to be influenced by the CEO, because I hadn't grasped the details of the plans (regardless of whether such plans really existed, as the important thing is if anyone thinks that there is something to grasp) or because I exposed weaknesses in the plans that were made by the CPO.

Although the incident weakened my ties to the French group, it strengthened the ones to the CEO. We kept working most of the night. Given that we had the undivided attention of the CEO, we could discuss the requirements until I was aware of every detail, even though I didn't agree that all of them made sense from a product perspective. Within a few minutes, I had drafted the technical design on a whiteboard, and then spent some time explaining them. They were complex and fragile from a database point of view, and the CEO wanted to explore all the possible alternatives, but I could prove that this design was the optimal one, despite its flaws. He finally agreed, and my team started the implementation of it. At 6:30 a.m., I had to get home to get a few hours of sleep, but then kept working during Sunday. By Sunday evening, the CEO was so happy with our progress that he ordered fruits and sushi to be delivered to the office, and he asked us to name what we wanted him to get for us. When I returned home right before 6 a.m. virtually everything was finished. There were still a few kinks and bugs around, as can be expected when building something in such a rush, but the basic functionality was there and the smaller problems could be ironed out over time.

Later, the CEO admitted that we had in two days built something that he didn't think could have been done in two months. From that point, we never again had the same kind of confrontation as we had had during those evenings. Before that, he had seen us fail because there was no clear way of knowing what the goal was or what we were measured on. But having worked closely with us, and seen how efficiently we could move from idea to design and then to implementation, I believe that he started looking for problems exclusively outside of my team (even if that included how we communicated with other teams). This occurred in the end of November, almost ten months into my employment. While I was happy that I had finally proved not only my logical skills but also my ability to lead my team, it had taken me far too long to do that. The explanation for that is something I'll have to search for within my own actions and behavior. Also, although his opinions about me and my team had changed at that point, it mostly affected how he perceived later events. Even at the very end of my employment, the critique in the review of my work was still completely focused on events predating the ones described above. I can't explain how the CEO could maintain two distinctly different sets of opinions, but unless he believed that our efficiency had changed radically it seems likely that some either individual or organizational mechanism was in place, which prevented reassessment of the old line of reasoning.

The CEO once told me that my way of arguing confuses people. He had realized that before even starting to discuss something, I already had 90 percent of the problem solved in my head. That means, that when I started to discuss I was only interested in solving the ten percent remaining, but only hearing about uncertainties makes people doubt that there is any substance in my claims. It was one of those statements that you don't know if you should take as criticism or praise. On the one hand, he was telling me about commu-

nication issues. On the other hand, he acknowledged my skills in problem solving. In either case, it could explain why basically everyone working close to me trusted me, but some of the others questioned my abilities.

4.5 Means to an end?

Given the product focus of the company, I would have assumed that the processes should have been streamlined to maximize efficiency. That's why I was astonished to see the amount of waste. It wasn't even always that the result of a fraction of the effort was used, since entire projects were just canceled without ever having been deployed, something which is presumably very risky for a company under constant pressure from shareholders to deliver. It was as if people believed that any action would bring us closer to the goal, even if there wasn't any thought behind it. Sometimes, however, it only brought us in circles.

The product actually changed its entire direction several times during my employment. This meant that every team had to change what they were doing, but not every front-end team changed quickly. Thus, back-end had to support different versions at the same time, something which created a huge pressure on us. These complete redefinitions of what the product was about started either as an internal initiative, or as a reaction to what our competitors were doing.

I promoted a more goal-oriented approach to the CEO, but he responded that what we were doing is called a *pivot*, and that it's something that many companies (especially start-ups) do nowadays. His point was that as soon as an optimal solution has been identified, different from what we've been aiming for, a course correction is the only reasonable action. Personally, I argued that it's easier to tell if a course correction is needed once the product has been launched, and that the primary focus should be on getting something out there. After all, doing "a pivot" every month is probably one of the most reliable ways to make sure that no progress is ever made.

Apart from the moving target, effort was also spent in organizational restructuring, which was later aborted without a formal decision. The CEO concluded, quite correctly, that there existed an educational need among the developers, especially on the back-end side. I designed a course in computational complexity theory, data structures and algorithms, with one weekly two-hour lecture. I estimated that I needed to spend one day a week on preparations for the lecture. However, I had only held two lectures before it was decided that these lectures should be postponed until later, when people would have more time on their hands. Since such a time is unlikely to occur in a privately held company, postponing the third lecture in reality meant canceling the series, although no one openly admitted it.

The situation with the algorithm lectures was similar to the subsequent management training, which was mandatory for everyone down to one step below vice president. Although the training was carefully planned—and the book *Good to Great* by James C. Collins handed out to everyone as required reading—it was obvious that not everyone had a positive attitude towards it. I even saw some vice presidents who displayed their unwillingness to participate through their body language in the conference room. Maybe partly because of the lack of support, but likely primarily because of changes in priorities, the third lecture was first postponed for one week, and then for another. I was the only one who showed up on the scheduled time the third week, and when I started asking around, people seemed to have forgotten about it. An e-mail message was sent out, stating that a new date would be announced shortly. This never happened, and no one mentioned the management training again.

Something that was apparently blamed for the lacking results was the incompetence of the developers. Now, while I could agree that there was a need for education, the lack of specifications made it impossible to actually trace the source of the problem. I perceived it as an organizational problem, and blaming the people at the bottom of the hierarchy just seemed like a convenient way to avoid fixing the real problem. One of the things I learned even before joining the company was that they had recently fired an underperforming em-

ployee and that they were now cleaning out his code from the codebase, and it was implied that he was the source of the problems. I also suspected that such rhetoric was used in the communication with the shareholders.

Although we never had anyone who was formally in charge of human resources, the task was handled by the CDO. As a senior engineer, I was often sent résumés and programming tests to evaluate candidates. After a while, I started asking around to find out who decided the recruitment needs, since some of these people seemed to be meant for my team, and I knew that I had certainly not asked to have my team expanded. It was especially strange to recruit more people at that time, because there were talks about not renewing the contracts for two of our current employees because of a lack of money. The CDO explained to me that his goal was to recruit the best people he could find, and then fire the worst people that we already had in our staff. To me, it seemed bizarre to spend so much effort on recruiting without a clear target. How would we know what to look for? My opinion was that an employee who could always solve the tasks that were assigned to him was "good enough," and shouldn't constantly need to look over his shoulder to see if anyone was measuring his performance. I explained that if you'd lock up the ten most clever people in the world in a room without giving them any instructions, it's of course possible that they will come out of the room a year later with a finished IT project that works exactly the way you want it to. But they would equally likely come out with a fusion reactor or the world's best mousetrap. Or, which is probably far more likely, they would come out with ten different projects because they never agreed on what to focus on. The CDO listened to my story, but it didn't change his opinion on recruitment.

The problems with recruitment were accentuated when a longtime employee was thought to underachieve. He had been moved to work directly under the CDO, who wasn't content with having to ask Tech when he needed someone to write code for him. But when the CDO was later unhappy with the employee's performance, he asked me to make a decision about whether the guy should be fired or not. He wanted me to review the work the guy had performed, and based upon that give my verdict. I refused to do it, since I couldn't fairly judge someone's achievements that they had made while taking orders from someone else than me. I hadn't even been in charge of this guy in the past. Since the CDO was not only the direct boss of the employee in question, but also responsible for human resources in the entire company, I felt that it was unfair to move this tough decision to someone in another department. In the end, the situation was extremely poorly handled. The employee had returned to Europe for Christmas. He had already arranged with a new Chinese visa and was even under the impression that he would get a salary increase upon returning to China, when he was informed at the last minute that he didn't need to return anymore.

I didn't understand the recruitment bonus either. It was a cash bonus that should be paid to an employee who used his network to find someone that the company eventually hired. But there seemed to be an official policy of only paying someone who actually knew about the bonus. For example, I recommended a friend who got hired as the first thing I did for the company, before hearing about the bonus, and never received it. Nevertheless, my understanding is that the CDO always received a bonus for recruiting someone, even if the recommendation came from someone else and he just performed the interview. Maybe there's a good reason for this, but it just seems to me like an incentive to hire people who were unlikely to stay in the company for long, leaving a position open for someone else who could generate a bonus for the one conducting the interview.

Another thing I found weird, given that our ultimate goal should be to deliver a product and make money, was the fixation on how the office should look. Many of the influential people in the company were fans of Apple products. This might be the explanation for why the desks were of the same kind as the display tables in Apple Stores. Their height was nonadjustable and because of their thick tabletop, the armrests of an

office chair wouldn't go underneath. Unfortunately, this caused one of my team members to have constant problems with his back, and some days he couldn't even come in to the office. If the focus had been on productivity instead of aesthetics, I'm sure the tables would have been replaced.

One Friday, a shipment of new computers and screens arrived at the office. Our old computers were very slow and the screens didn't have very high resolution, so a switch was certainly called for. The order was to wait until the end of the workday with unpacking the new equipment. By the time we got started the CPO had already left the office and didn't know what was going on. Everyone got two screens, and we were a few people who rotated one of our screens from landscape to portrait, to make long code listings and such easier to read. I was at the office during Saturday as well, and in the early afternoon the CPO and some of his French friends arrived. The CPO looked around the office, and then turned to his friends and spoke in an agitated voice. Then he turned to me. "Whose fault is this?" he asked. "What?" I responded, not knowing if he was actually serious. "Who was the one who decided to rotate the screens?" he clarified. "I was," I told him, although it wasn't true. Someone else was the first one to do it, but I had the same idea and I didn't see the problem or care about who started it, and figured that it was better if the CPO was angry with me than him. He picked up a stuffed toy and threw it at me as hard has he could. "You've ruined everything. I ordered the new equipment because I wanted the office to look good. But just look at this! Three people have already followed your example!" He was looking around the office as there were rotated screens in several places. Not being completely sure how to tackle the situation, I tried to joke and said "I made it look like a skyline." He responded coldly "That's not helping" and left. At that point, I was really close to resigning. I couldn't imagine how it could be more important that the screens were all at the same level, than that the people who would spend their days working with them could freely choose the configuration that would suit them best.

It surprised me how often the company could identify a healthy strategy, and then fail at its implementation. During my final month of employment, the company management informed us that as part of their marketing strategy, they would write a book about fashion and how it had developed in China over the last few years. While I personally had nothing against this idea—I believe there were a few who had, however—I was a bit puzzled when rumors reached me after I had left that the one assigned to writing the book was the CDO himself. Now, I have no reason to question his skills as a writer, but as far as I know his only qualification in fashion was that he started at the company as a photographer. The way I see it, a book about car design should discuss how the design is not only aesthetically beautiful but also adds to fuel economy and safety. Likewise, a book about art should give the reader factoids about the artists' lives, different techniques for painting and how opinions about art have changed over the course of centuries. Despite the fact that the CDO had a few million photos at his disposal, I don't see how the kind of insights that typically make a book interesting to a reader could be gained simply by superficially going through the photos. This particular decision interested me. I have to assume that there is information that isn't available to me, but my inquiries were met with snappy responses. For example, I've heard rumors that the book will contain more pictures than text, and that there is someone with an education in fashion who can help out, but the CDO is still the main author and I don't think that the quality of a short book should be expected to be worse than that of a long one.

4.6 Organizational mythology

Organizational myths are important in shaping a culture. Hence, I paid attention to potential driving forces for creating myths in my company.

During my entire time at the company, they called themselves a "start-up." Given that the company had been around since 2007, I would argue that they should have moved past the start-up phase, but I was wondering why it was so im-

portant for the company to classify itself in that way. Is it that six years isn't considered such a long time, at least in China? Or could it be simply that the company had reinvented its business and was focused on a new project? Another interpretation could be that the company tried to maintain a start-up culture, by not focusing on building structures. Yet another is that there are certain benefits associated with being a start-up, for example an excuse for paying low salaries or a set of buzzwords for speaking with potential investors. Regardless of the reason, no one I talked to seemed to question the start-up classification.

In June 2012, the CEO asked me how I would solve a network communication issue between the iOS app and our servers. This wasn't anything that I was normally working with, since the mobile team usually dealt with the app and Operations took care of the servers, but it wasn't very difficult for me to create an action plan, which I then presented to the teams involved. I made sure that someone started to investigate according to my plan and after about a month I got an e-mail message that the issue had been handled. I checked with the mobile team if they knew more, and got the answer that they hadn't heard anything more about it and would await bug reports before spending more effort on it. With that, I assumed that my responsibility was over.

Much later, the CEO asked me how work with the mobile app issue was going. I told him that I was under the impression that it had been resolved, and that the mobile team wouldn't do anything unless they were told that the issue still remained. The CEO said that I couldn't assume that the mobile team could handle it on their own, and that I should work actively to verify that the issue had been resolved. I explained that trying the app with different content, different mobile network providers and from different locations in the city is a time-consuming task, and I asked if he wanted to make that my responsibility. He replied that he wanted everyone in the company to feel responsible for everything that the company did, since that would ensure that as soon as anyone spotted a problem, it would immediately get attention.

The idea is interesting, but I don't see how it would work in real life. Assigning responsibility to everyone at once is the same as not assigning it to anyone at all, since you can't punish the entire company for every single mistake. Also, it's unlikely that everyone is as good at solving every problem. Besides, such a structure makes it difficult to know how to prioritize and distribute work since, as a result, everyone is working on everything. The mental image I got was of a youth soccer team where everyone, including the goalkeeper, are running for the ball at once. But regardless of whether this idea would work or not, it describes an important aspect of what the company was striving for.

Another time, I was telling the CEO that it would help me to have a more detailed description of hierarchies, responsibility and accountability within the department. He responded that the assumption that we needed to be defensive about our work was a lousy one, since he wanted people to be able to work together as a team towards a common goal, and that he wanted the organization to be as flat as possible. While I naturally agreed that we needed better teamwork, I had to accept the fact that there were severe problems within the organization. Instead of hoping for things to get better on their own, I wanted to enforce stricter rules, at least until we were back on track. On the other hand, there are often many ways to achieve the same result, and there's no reason to believe that we couldn't do it with a flat organization as long as the problems were dealt with in some other way.

It's also interesting what role reputation plays in an organization. Especially within the French group, myths were created around people because of the companies they had worked for or received offers from. It doesn't really matter if these myths were true or not—some of them were notoriously difficult to verify—as they served their purpose of uniting people's opinions both inside and outside the group. On the other hand, such myths make some skills seem magical, for example the CPO's design skills. While I'm not in a position to know, since I quite simply never saw enough of his designs, I know that others were

impressed. Still, I believe that design is just another skill that you can study and learn, and although some people learn it quicker than others, I don't believe that certain people have the gift and others don't.

For some reason, I found myself in a long discussion with the CEO about design. I was telling him about design principles such as *area alignment*, the *golden ratio* and *closure*, in other words principles that work because of how our senses operate, how the nature around us is shaped and our need to find patterns in incomplete or ambiguous information. To my surprise, the CEO strongly rejected the idea that such principles exist. He simply wouldn't accept that leading designers could agree on a list of "facts." By choosing that approach, he inadvertently elevated the skill of design to something that comes from deep within ourselves and that can't be taught. Another possible conclusion is that good design can't be evaluated through anything but a feeling. I've been wondering if such a myth makes people trust others who claim to know something about design more, and if it could possibly impede the recruitment of more designers to the company.

4.7 Inefficient communication

When I started to work for the company, the API that was in place had mostly been designed ad hoc by a bunch of individual developers. It was built on a somewhat robust framework that had once upon a time served its purpose, but had since then been misunderstood by most developers, beyond the fact that it lacked the flexibility to handle many of the requirements in the increasingly complicated system that was being built. I had been of the opinion that this API needed to be replaced, and that the codebase that was supporting it should be rebuilt at the same time. But due to the high pace at which new features had to be implemented, the plans were put on hold as everyone involved understood that the process would be extremely costly.

In the end, however, the decision was made for us as we ran into a stone wall. The company urgently needed an Android app built, and since we didn't have any Android developers on the team at the time, the task was outsourced to a company in France. A meeting was held on a Thursday afternoon, where the CPO said "We sent information about the API to the developers in France, but they refused to work with us unless we can provide something better. You've been speaking about this new API that you want to build. How much can you have done by Monday?"

The question was of course preposterous. The back-end for a company in our industry can't be designed and built over a long weekend, even if the product would've been less complex than ours. I thought for a while and, given the extreme pressure that was now put on me, gave an estimate that I shouldn't have given. I told them that, under ideal circumstances, we should be able to design the API in three weeks, and then in another three weeks build a thin layer that would be used to convert data between the old crappy format and our new clean format, but not do any deep rewriting at all. The CEO answered "But I know that what can be done in three weeks, can also be done in two," and all of a sudden I had four weeks instead of the six that I had requested as the bare minimum. We agreed on a set of objects that the new API should expose, and given that the shareholders had been promised an Android app by a certain date, I felt like I didn't have any other options than to try to accomplish it.

Of course, the conditions didn't turn out as ideal as I had hoped. The first thing that happened was that one guy was temporarily moved from my team to work on another project, and he stayed with that project for the majority of the time. Then people took turns being ill. We were of course delayed, but except for one or two days, we made it within the six weeks that I had originally estimated as the quickest possible time frame in which we could accomplish anything. I saw it as a success, but others that had expected us to do it in four weeks saw it as a failure. This was just one of many misunderstandings that would occur during the development of the new API.

Over the following months, the requirements for the API changed continuously, and we did our best to keep up. This meant that we could

never reach a state where we felt that we were done. Worse yet, since one of the responsibilities of the API was to represent the relationships between objects, one change "here" would often cause three changes "over there." We were asked to communicate more about what we were doing, and every time we were asked this we created and shared documents explaining the current state of the development. We spent a lot of time writing these documents, but they were rarely, if ever, read by anyone. Internally, we felt like we were shooting at a moving target. Externally, I'm sure it seemed like we didn't know what we were doing since everything changed all the time. We constantly asked for feedback on our proposals for new features, but far from everything was answered. More than one person complimented me on the comprehensive e-mail I sent, but at the same time regretted that only a fraction of the messages were responded to or acknowledged. Later I found out that the consumers of the API had mostly been sitting on their hands, getting more and more annoyed that we were never done. What would have helped us the most was timely information, but the lead mobile developer saw the changes that were going on and decided to wait until everything had been finished, since his main objective was to construct an app, and didn't want to deal with an API that was constantly changing. The lead web developer had even less interaction with us, as he was still working with the old API and didn't have plans to move to the new one anytime soon. This meant that we usually got our feedback only once a feature had been designed, built and released for internal development. It caused a much more costly change process, and chances were that the final result wouldn't even be as good as if it had been built right from the beginning.

Unfortunately, the lack of feedback sometimes made us forget about implementation details that we had proposed. I wanted to wait with the final decisions until I got approval, and waiting for them I switched focus to other things. As no approval arrived and no decision had been made, the features had not made it to our backlog yet, with the result that some features were delayed or even completely overlooked. Because of this, I eventually altered the workflow so that every proposal e-mail message contained a response deadline and section about what our course of action would be if no one replied. This did little to limit the amount of criticism we got for not delivering what was requested, but at least it improved our efficiency and prevented us from being blocked by the lack of feedback. On the other hand, it made us jump to conclusions, knowing that the information we had was far from enough. In an extreme case, I even apologized to a team member for asking him to build a feature that we knew would have to be rebuilt in a matter of weeks. But it was at the time more important to quickly deliver something regardless of its quality, as we would often get our feedback only once a certain feature had been completed.

Communication was hardly improved by the intermittent fights between my team members and people outside the team. Because of misunderstandings or unnecessarily harsh words or sarcasm, people were offended to the point where they lost their motivation to work, and others could even need to intervene to reach a truce. I lost countless hours of productivity within my team, and it even resulted in rules about who was allowed to send e-mail to whom.

At one point, after several months of added requirements, constant redesigns and criticism for being extremely overdue, a meeting was held to ensure that the basic features were finalized so that a new version of the iOS app could be released. I made a list of the objects that needed work, and the time frame was once again optimistic: everything would have to be completed within two weeks. My conditions for accepting the task were that I would be given the undivided attention of the iOS team, to ensure that our result would be something that they could easily use, and that I would not have to spend time writing a plan for when different parts would be done. Instead, I wanted the freedom to move between objects as I saw fit, and since the objects had a complex system of dependencies between each other I wanted to develop many of them in parallel. We would have daily progress meetings, and after the

two weeks the back-end team would give the iOS team the same kind of attention to help them in their work.

The cooperation between back-end and mobile during this endeavor was surprisingly smooth, at least given our previous differences. But then the CPO told me that he would assign Frank to me, to help me with "time management." Frank, who had not participated in the initial meeting and thus didn't know what had been decided, hosted the daily morning meetings and immediately started requiring a detailed plan for when different objects would be completed. I was upset, because the person that I had been told was supposed to help me instead became an informal boss that I had to report to and spend time satisfying with documentation that I had initially been promised that I wouldn't have to provide. But Frank's opinion was that nothing could be guaranteed to be accomplished unless it followed a plan. Hence, I reluctantly agreed to create a spreadsheet which I kept updated with the current state of all of the objects. However, unexpected changes could cause entire groups of objects to be moved back from *done* to *in progress*. This is not surprising given the nature of our work, but Frank wasn't prepared for this. He wanted to be able to count the number of red rows in the spreadsheet and measure how quickly they turned to green, and what he perceived as regression made him frustrated. Several times, he also asked me to provide a list for the mobile team of all the requests that were possible to send to the new API. To me, this demonstrated a lacking understanding of the new structure and that Frank was trapped in the thinking of the old API, which was the only one that he had actually worked with. The new API was designed to be flexible enough to answer any reasonable query. Asking for all the possible requests made about as much sense as asking for a list of all the words that the Latin alphabet could be used for. I turned to the lead mobile developer, who was also present in the room, and asked if this was really necessary. He answered that he understood the structure without further explanation, and that he could do fine without such lists. He also agreed with me that our communication had recently been adequate, and that we didn't need outside help to make it work. Frank then asked if a particular detail in our implementation was currently working, and I responded that I'd have to investigate it and get back to him with an answer. My belief was that he had identified a current problem, but upon investigation it turned out that that feature had never been broken, and I got back to Frank with an instruction on exactly where in the documentation, which had been around for months, we had explained this feature. This made Frank furious, and the issue got out of proportion and involved other people too. In Frank's opinion, my job was to be able to provide him with exact information about the current state of every part of our system, regardless of how deeply I was involved in the work on the new parts. He had no responsibility to learn about our system, and his question was only to test me and see if I had prepared well enough.

Despite the problems that we had had, Frank continued to be the one that prioritized the tasks for back-end even afterwards. I was very proud of my team, which worked very efficiently, and internally we all knew that what we delivered was what was asked of us. Externally, however, we continued to receive critique for not building what had been requested. The CPO eventually sat down with me to find the cause of the problems. I could show him the lists that Frank had sent me, and could match them with our results. I could tell from the CPO's reaction that he understood why it had been difficult for us to meet any requirements, and he admitted that the bullet points, which were usually shorter than a sentence and often just contained a single word, were nowhere near sufficient specifications to guide a team of developers for a week at a time. I forwarded a number of such e-mail messages to the CPO to have a look at. He brought it up with Frank, who apparently couldn't see the problem. In his view, everything that was present in the old API should simply be developed in the new, and that was why he didn't put more effort into explaining to me what he wanted. But that idea was absurd to me. There was so much redundancy

and so many incomprehensible fixes in the old API, that what we already had was beyond repair and we were forced to start over from scratch. After all, if the old API really was ideal, then there wouldn't have been any need to create a new one. I felt like I had been handed an 18th century sextant, and was expected to use it as a model for building a GPS receiver.

4.8 Lack of specifications

When I joined the company, the team was claiming to run Scrum but, according to my experience, it was a misinterpretation of Scrum. Frank, who was the Scrum Master, joined sides with the CPO, who was the Product Owner, and pushed the team to accept as many tasks as possible. Instead of the usual method of assigning points to the tasks by everyone simultaneously, the authority on the task was asked to estimate first. As other people estimated, few would increase their time estimations from what has been said earlier, which means that in the end, the team almost always ended up with a far too optimistic estimate. At the end of the sprint, most features could usually be demonstrated, but they were rarely good enough to be released to production. The problem of *always* building *almost* complete features seemed to surprise everyone. But given the Pareto principle[10], it's exactly what's to be anticipated. If each task is assigned 20 percent of the required time, then only 80 percent of the expected functionality will be implemented.

After these nearly-completed tasks had been demonstrated at the end of the sprint, they were surprisingly often abandoned. They were either removed or, even worse, left behind in the codebase as a vermiform appendix—a vestigial remnant of a buggy deprecated feature that is likely to only cause a disease sooner or later. To me, this was a paradox. How could a feature that was a top priority just a few weeks earlier and had nearly been completed suddenly be discarded? One explanation that I heard for certain features was that their implementations were so messy that they couldn't be salvaged. Another explanation was that Scrum require you to reassess the goal of the product and reprioritize its tasks between each sprint.

I accepted neither of the explanations, however. A feature should either be important enough to be more or less a requirement for the product to succeed, and then it should be developed even if the cost turns out to be higher than initially estimated. Otherwise, it was never important to begin with, but then it should never have been designed in the first place. Sure, the situation can change over night, and the company might have to reassess its entire business idea, but slight changes in trends are to be expected over the course of the development of a complicated product. The design of a new product should preferably be a conscious action to pursue a vision, not a sequence of reactions to what other companies are doing this particular week. Chances are that such a strategy will only lead in circles. How could you reach your goal if all you do is turning your motion in new directions?

I believe that the real reason for the behavior I observed was the misunderstanding that Scrum encourages you to redefine your goal every few weeks. But Scrum is more about not biting off bigger pieces than you can chew. By having a clearly defined end result, but prioritizing the steps to reach there and every few weeks compiling everything into a theoretically useful product, Scrum allows you to check the sanity of your idea and correct any mistakes before things have gone too far.

After I had pointed out the weaknesses I had seen, some small improvements were made, but it wasn't until I got in control of back-end that I really started to push for specifications. Since many of the important positions in the Product department were held by Frenchmen (product owner/ designer, lead mobile developer and lead web developer), they were the ones I had to speak to. But holding long meetings was not okay with them. Discussing any kinds of details was not

[10] The *Pareto principle* or the *80–20 rule* claims that about 80 percent of the effects come from 20 percent of the causes, a principles which is observable in many large systems. For example, about 80 percent of the world's income is controlled by the 20 percent richest.

okay. Sending too much e-mail or using IRC[11] was not okay. And neither was speaking about anything work-related during lunch. The CPO explained at an early stage that I didn't belong to the target group, which efficiently ruled out my own thoughts and values as a way to reach conclusions, and yet no written specifications could be produced. Once, after having struggled for a few weeks trying to grasp one of the new concepts that we were about to implement, I got to have a meeting with the CPO where I could ask any questions I wanted. I was sitting the day before the meeting, writing questions and follow-up questions that depended on what his answers would be, and then ran them by the CEO in the evening. The meeting was very informative, and I got a much better understanding although I primarily got to see the graphical design, and didn't learn much about what the goal of the feature was or how it fit into the larger picture. Still, it was light years ahead of what I had been given before. But I received criticism several times afterwards for taking too long, in a meeting where everyone but me was claimed to already understand.

What I primarily wanted to know was what the user should experience when using a certain feature. With that knowledge, I could deduct the details myself. But the information I was handed was at a much lower level, often even at an implementation level that didn't even make sense. The only thing I could do with such information, not knowing anything about the larger picture, was to implement it in exact accordance with what it said. Afterwards, it would of course turn out that the result didn't meet the expectations.

The CEO had a theory, that as long as I couldn't deliver what the Frenchmen wanted, they felt that they couldn't give me the overall picture. And each time that I failed to meet their expectations, they gave me even more meaningless implementation details and even less of what I really needed, essentially trying to do my job for me. This would quickly become a vicious cycle, in every loop reinforcing their resentment towards me and my frustration towards them.

[11] *Internet Relay Chat*, a protocol for group chats over a network

But although there could be some merit to this theory, I believe the main explanation is simpler. I don't believe there was a definite goal. The idea was to release an app and make money, not to serve a certain purpose or satisfy a certain user need. Without a definite goal, specifications describing the overall picture couldn't exist, and with no specific problem to solve or needs to meet, any feature would be as good as any other. This could explain why some features were never finished, and why that was accepted.

A number of months into my employment, the CEO had a conversation with me where he explained what he perceived as my weakness. What he said was that I didn't think in terms of user experience, something I disagreed with. I started pointing out inconsistency after inconsistency, such as where buttons should go and how the flow through the interface should work. He replied "This is what I'm talking about. You focus on the details, but what I want to know is what you think the app should do." I was dumbfounded by his request, but after having thought for a while I gave him some quick ideas. He was equally surprised by seeing me coming up with ideas, but the truth is that I never thought that it was my place to suggest ideas that would change the direction of a product that had been developed since long before I joined the company.

Once, when I had pushed the CPO to give me specifications for a feature that I was about to build, he responded "The reason I haven't given you any specifications, is that I don't know what you'll be able to do." This is of course a strange way of reasoning. The logical way would be to describe in detail how something should ideally work, to allow for a discussion about the feasibility of such a plan and possible alterations to limit the required resources. He continued "I've also refrained from giving you too much information, since information would make you narrow-minded. I want a solution that is as generalized as possible." His way of arguing was completely unfamiliar to me. If I had been asked to construct a building, I would have wanted to know if I was supposed to build a small shed or a factory. Asking for a generalized solution doesn't always

make sense. But it would make sense if it's the result of missing a long-term goal. It still remains to be explained why we were constantly blamed for building the wrong thing. However, maybe the plan to find the correct implementation was "I'll know it when I see it."

I have been theorizing about two forces that might have been at play, that could possibly explain parts of the behavior I could observe. The first of these stems from the business the company was in. The company had several investors and shareholders, and the most important performance indicators were based on the total number of users and the number of active users. In those terms, it might seem counterintuitive to limit the pool of people that we could hope to attract as users by explicitly defining who they are and which problem we are solving for them. With that mindset, it could make sense to attempt to be everything to everyone. In other words, if anyone would reject us for any reason, it would be seen as a failure. This would explain the massive amount of features that we implemented before releasing anything and why the question about what the app should do was open for everyone to contribute to. And indeed, the CEO even told me at some point that his ambition was an app that would replace all other apps.

The second possible force is the fear of failing. As long as the product is still under development, everyone can rest comfortably in the position they're in (at least until the company runs out of money). You can continue to do the same thing you've been doing for some time, and you can retain the prestige associated with the brand and your position. But releasing and actually seeing how the product would succeed in real life would be to open a door to the unknown. If it's a hit, your workload might increase rapidly. If it's a complete failure, you might be out of a job in a matter of weeks. In either case, finishing the product would change things drastically, and I'm not convinced enough people really wanted that to happen. On a smaller scale, this could also be about personal failure. To illustrate, by writing a solid specification you could be held accountable for whatever mistakes or miscalculations are in there. But by pointing in a general direction and waiting for other people to do the job for you, any mistakes made would be on their part. If you're at least skilled enough to tell whether a feature works or not, then you could always blame others for not implementing what you asked for if it doesn't, and get the credit for your awesome design and management skills if it does.

Admittedly, things gradually changed for the better. I'm not sure whether it was the result of the continuous pressure from me to produce tangible specifications that I could follow, or because the recruitment policies had changed so that we hired more skilled people, causing a change in attitude in the office. Probably it was a combination of the two. As I realized that it was difficult to get the specifications I wanted, I started requesting *personas* instead. Personas are a handful (preferably not more than three) of descriptions of target users. They could be made up, but might be even more interesting if they're real people with information collected during interviews. By learning about their age, education, interests, families, favorite quotes and goals in life, you could almost imagine that you know them. When using an actual person as the basis for your discussion, it's easier to agree on which features they would use and not. With personas as the starting point, I was hoping to be able to deduct most things myself, with just a few keywords to guide me. Furthermore, I argued that the personas would not only help to communicate what we were trying to achieve, but also act as sanity checks to make sure that we're on track, while improving shareholder and advertiser communications by building *use cases* on the personas, which could be compared with the actual features. They would allow us to see when we had reached our primary objectives, and the advertisers could get a vivid image of whom they would be able to reach. Unfortunately, it turned out to be difficult to get the personas developed. When I suggested it to the CPO, his only reaction was "I wish you had suggested this earlier."

What was slowly being accepted, however, was the use cases that would normally have been developed as the next step after the personas. A use

case describes a situation where the user would have a certain need, and uses the product to accomplish their objective. The first iteration of use cases contained little more than mockup[12] screenshots and lists of options that the user had, but at least it was way better than not having anything at all. Once a tangible document was in place, it opened up for comments, discussions and improvements. These documents were not created by the CPO himself, though, but by a series of technical writers who were one at a time assigned the sole responsibility to create these use cases. While the information asymmetry was still present and caused some frustration, it could at least be handled by someone working full-time with only documentation and requirements.

4.9 Expectations and prejudices

When I first started to work for the company, I was hoping to become friends with everyone in the department. There was especially a group of Frenchmen, who always had lunch together, but I was quickly accepted into that group and joined them at least every other day. While the lunches were fairly exclusive, a lot of people also went out for coffee at around 4 p.m., including people outside of the French group.

But having returned from a trip, I noticed how things had changed. I was no longer invited to lunch, and the daily afternoon break had also become exclusive for the French group. The way I saw it, this was an unfortunate but unintended development, and one time when I saw them leaving I followed them and entered the elevator just as the doors were closing. The CPO told me "We're not going for coffee." I answered that it didn't matter, but the CPO repeating what he had just said. I maintained that it didn't matter, and I would go with them to do whatever they were doing. Indeed, the CPO was right and instead of going for coffee they were planning to have chocolate with whipped cream, but the atmosphere was somewhat awkward as I knew that I wasn't welcome. When the CPO later wanted to speak with me about how to mend the deteriorating mood at the office, I brought this up as an example. His response was that he remembered the occasion, but that it was a special situation because he needed to speak privately to someone. This explanation didn't really make sense, however, since the entire French group was present.

When I speak about the French group, I refer to a group of people whose salaries were relatively high and who always had lunch at comparatively expensive restaurants. Apart from them, there were also a few French employees who were never invited by the others. They didn't seem to be appreciated in the same way, and their salaries were comparatively lower. But there was also a clear distinction in their attitudes, and they didn't make a difference between nationalities. While speaking with the CPO about how to improve relations in the office, the CPO pointed out that the problem also lies with others, and gave me the example of when one of these other French guys had once asked him "Why do you never have lunch with me?" The CPO had asked him "When have you ever invited me to lunch?" After that, the guy never asked the CPO again. But to me, this was clearly splitting hairs. As correct as the CPO's answer may have been, the original question was in fact meant to be a lunch invitation, and the answer was a rejection of the request. If the CPO had wanted to, he could obviously have chosen to interpret the question in another way. What confused me, though, was why he would use this as a demonstration of why other people were not doing enough to promote team-building.

An explanation to my problems with the French group I got much later from the CEO, was that I had had a good relationship with the head of Operations who was working there when I started. The French group didn't know how he was working and didn't see the results that they were expecting. Hence, they assumed that he didn't do his job and refused to even speak with him, and because of guilt by association the same must have been true for me and I ended up being treated the same way. It seemed to me that they had originally collectively decided on him being the problem, something that then transferred to me as he left the company.

[12] a model of a design, often a design prototype

Later on, I observed how the collective blame was directed at the CEO instead, and he was accused of impairing the development by micromanaging. There was an attempt to enforce the structure of the Product department by making sure to always follow the proper chain of command, and direct all outside requests to a single gatekeeper. This attempt seemingly failed as the efforts to isolate the department ceased (and possibly partly because of the incident described on page 23), and the blame was then once again directed at me.

I don't know how much the blame was a result of actual criticism—it was difficult to get any feedback at all—and how much it was a result of the organizational structure and other possible problems that could be difficult to distinguish. After all, as the head of back-end I was responsible for the infrastructure that both our web and mobile front-ends needed to access. As long as I didn't do my job properly, they couldn't move forward either, and in the managerial positions responsible for these teams there were Frenchmen who had known each other since before they started to work for the company. Sometimes, I even got the impression that their opinions weren't their own, but that they collectively decided what to think. For example, when designing the API one of my principles was to make the responses easily interpretable even on a mobile device with limited computational and storage capabilities. That's why I was surprised when I was approached by the lead web developer and told that the design was poor. Later on, during a meeting, I asked specific questions to the lead mobile developer about the design, and he didn't have any problems with it. This apparently made the lead web developer change his mind—something I didn't hear from him directly, though, but from someone else. His reason for criticizing the API was that he believed that the design didn't suit the mobile team, not that he had personally identified any problems with it.

During my time in the company, I interviewed several people for different engineering and Operations positions. One of them was a Frenchman who was interested in finding a job in China. It was through a recommendation from the lead mobile developer that we found him, and he did very well on the Skype interview that I held with him. He accepted our offer to work in my team, and I sent him information about the work that we had already done since he didn't have much else to do until the start of his employment, and he had asked for something to read. I appreciated that he responded with ideas for improvements, even though these ideas were dead-ends which we had already been down. But in subsequent e-mail messages he insisted on his ideas and didn't take my responses into account, until I finally was so annoyed that I wrote a short reply to postpone any further discussion until we could meet face-to-face. At this time, I already understood that he must have been told by the rest of the French group that back-end was a mess and that it would be up to him to make it work. His entire employment at the company was characterized by an unwillingness to follow the same rules as the rest of the team, criticism regarding the way we were working or what our codebase looked like (delivered through someone else who had heard it from some other French guy) and an inability to explain his ideas to the rest of us. At the same time, I was told several times by the CEO that it was very important that we could bring this guy in on how we were working, to improve the relations to the other Frenchmen at the company. Trapped in situations where the group work completely stalled because the entire team except for him agreed on the course of action, and we couldn't get him to explain why he thought our solution wouldn't work or what we should do instead, I made my probably biggest mistake. I put this guy in charge of developing things his own way, with the support from two other developers who were working under him. When he finally gave up and asked to be relieved of this responsibility (which he didn't do directly to me, but to the CEO), we had lost much precious time. In the end, company management decided to let him go. He probably thought that the decision was mine since he didn't say a word to me and immediately removed me from Skype, but I wasn't even informed until he was already gone. How

he could have performed so well on the interview but completely disrupt our team work was a mystery to me, but he must have been influenced by the French group into believing that anything anyone in the team said must be wrong.

It struck me how much expectations influence the interpretation of what we hear, when I learned how differently different people had interpreted Dan's words. Dan had told the CEO, months after the project started, that he finally saw what all the work with the API had resulted in. Although he hadn't appreciated the way we were working at first, he had later started to read up on other APIs out there, and realized that our solutions were better than most other things he could find. The CEO told me this, and I could hear the pride in his voice. Dan himself brought this up with me much later, and told me that he thought that the API was the greatest thing that the company had ever built, and that it was flexible enough to handle any future alteration. But when the CPO mentioned the same thing to me, his words were "I heard that not even Dan—one of you own team members—understood what you were doing until now," implying that what we had done was poorly documented and overly complicated. It must have been obvious even to the CPO that Dan, apart from being one of the earliest employees and hence the one most accustomed to the old way of doing things, had been moved around between projects and done very little that had anything to do with the new API. Yet, the CPO's prejudices about my work had made him interpret the same message in a diametrically opposed way.

In December 2012, the CPO stopped speaking with me completely. I had recently decided, together with the CEO and the interim CTO, that my new job title would be *Director of Engineering*. It made sense, because I was below C-level (vice president) but with a team large enough that it could be split into two, theoretically in that case led by two Managers of Engineering. Nevertheless, the CPO sent me an e-mail message telling me that the new title he had seen on LinkedIn was inappropriate, and asked me to change it. I knew that the CPO had wanted me to have the title *System Architect*, something that would have been an inaccurate description of the work I was doing. Firstly, I was managing the largest team of developers in the company, something an architect typically doesn't do. Secondly, the title would imply that I had designed the structure of the source code itself, although I hadn't even looked at it. I sent the CPO a bunch of links, describing what the different job titles meant and how a corporate hierarchy is normally built. He admitted that I made some good points, but asked me to hold on until he had cleared it with the company management. He never got back to me with a response. Instead, he quit talking to me altogether. While I can't know his motives, I suspect that he wanted to prevent me from ordering new business cards and introducing me as a director, probably because he didn't think I deserved it. In combination with his negative view of me, the fact that he had recently lost control over the developers, who were all moved from the Product department into the newly created Tech department, might have added to his frustration, especially since I had been arguing for months that such an organizational change was necessary. At approximately the same time as the CPO asked me to revert my job title, a parcel arrived at the office that Randal wanted me to have. But instead of letting me have it, the CPO took it and kept it for himself, the reason for which is still unknown to me, but my assumption is that it's connected with the other things happening at this time, maybe as a revenge. While no one knows for sure why the CPO stopped speaking to me, different people had different ideas. Thomas attempted to find out, but he said that the CPO's critique was unspecific and unprofessional.

Towards the end of my employment I was helping the iOS team, which needed assistance to quickly fix bugs and develop new features. But this was problematic, since the French developer who remained on the team had a negative attitude towards me. He blamed me already during the initial phase of this cooperation for a mistake that he had made, in a very derogatory tone. The CEO wanted me to demonstrate in detail what had really happened, something that occupied the next couple of hours for me. While it wasn't dif-

ficult for me and others to prove that the accusations were wrong, what made me feel bad was the attitude of my colleague, who seemed to be so sure that any mistakes must have been on my part that he wouldn't even listen to reason. Later on, I started to hear rumors that he had been complaining about me behind my back. He had said that I was incompetent since I hadn't solved as many bugs as he had. I'm sure that if he had given it some thought he would have realized that the bugs I was working on were generally much larger than his, not to mention the fact that I hadn't volunteered to work on the iOS team. It was a decision made by someone else since that project was in a state of crisis, and there wasn't much else to do. I had never written an iOS app, and I was certainly not used to the code that he and his former teammate had written. If he had thought about it, I'm sure he would have agreed that it would have been unfair if he instead had been moved to the back-end team against his will, and then been criticized if he didn't solve bugs as quickly as the people who actually designed and built the system. But to me, this was just another example of prejudices strong enough to prevent people from thinking clearly. I found it extremely difficult to work next to someone who despised me that much. I finally lost all motivation to work, and in the end it became my most important reason to resign.

4.10 Match fixing

I sometimes got the impression that things were presented to me in a way so that I couldn't use them to my advantage. It usually happened with the CPO. For instance, when accepting the job offer, the CEO told me among other things that I would get an iPhone so that I could use the company app. To me personally, it didn't really matter as I had bought one of the competitors' more expensive smartphones just a few months earlier and would be paying for it for almost two more years. A few weeks into my employment I had already forgotten about the iPhone, when the CPO asked me to have a word with him in private. He told me that he had heard that a promise had been given to me without his consent, that it was a bit early and that he wished he could have made the decision once I had presented some accomplishments. He then handed me a package which turned out to contain the smartphone and said that he was sure that I would be able to deliver value in the future, but asked me to keep quiet about it to the others as it was a bit unfair.

I used the phone for my Chinese SIM card and could use my Swedish SIM card in my other phone. While this proved to be convenient for me privately as well, I still ended up using the phone a lot in development. Receiving a tool that I needed to do my work—which I hadn't asked for and still been promised by someone else—together with a speech about how it's too early made me feel bad. My impression was that he wanted me to be indebted to him, and at first I was reluctant to even unwrap the phone.

An incident of a slightly different character was when I would negotiate my salary with the CPO (something I didn't even saw the need for as the CEO had already told me what my salary would be). The CPO told me that there was a wage ceiling, and that no one in the Product department earned more than that. He also told me that I would get that amount and that he was hoping that he could raise it when there was more money available, something which never happened though. We weren't allowed to discuss our salaries, but much later I heard from one of my colleagues—someone that I had interviewed and hence started to work much later—that he was also told about the maximum wage. Since it was only the salary itself that we were prevented to speak about, he could tell me about the maximum wage according to the CPO, and it turned out to be less than what I was earning. Later, I also heard a rumor from someone who had by mistake obtained access to some numbers, that the CPO himself earned almost twice what I was making.

Of course, you can argue that I shouldn't just have accepted the offer that was given to me. In Sweden, I would in fact have asked for more, but the different tax rates and costs of living, uncertainty of common Chinese levels of salary when

being on a local rather than an expat contract[13], promises of eventual employee stock options and the hopes of an increased salary within the nearest future made me choose to wait and see. However, once I started learning that different people were told different things, I realized that there was never supposed to be any negotiation at all, and that it was just a masquerade to give people what the management had decided. I don't know if this is the usual practice in China or France, but since I'm used to the Swedish way of arguing about the strengths, weaknesses and potential contributions I felt a bit tricked when thinking about my salary. Even though the salary shouldn't be the only motivation to work, it's one of the most visible parts and the ongoing feeling that I had been tricked added to my feelings of dissatisfaction.

There was a range of other situations, where promises were made but never honored. These were promises about vacation after working overtime, or other kinds of compensations and arrangements such as my requirement for accepting the job offer that I'd be allowed enough time off work to write my business thesis already during my first spring. In the majority of these situations, however, I feel confident that the outcome wasn't planned. Instead, a mix of problems with priorities, myopia and a strong desire to immediately respond to shareholder requests made the plans change, in a way similar to the canceled corporate education programs (see page 25).

4.11 Undervaluing the talented employees

The first time I met Randal was during a corporate outing to the Commune by the Great Wall in April 2012. At first I had no idea who he was, but a conversation about photography with one of my colleagues soon involved him as well. Randal joined my side in the discussion, and made thoughtful remarks. When we had lunch, I explained esoteric programming languages to another colleague, and Randal overheard the conversation and enthusiastically responded. Esoteric languages are often created with the sole purpose of being geeky, difficult to read and allowing clever programmers to show off. At that point, I understood that Randal had a profound knowledge of and love for programming. No one learns about such languages without being skilled in at least a few regular languages, and no one spends their free time (because this isn't something you get paid to do) on something they don't truly enjoy.

I learned that Randal had been interviewed for a position with us, but that he had also spoken with Microsoft. After the outing, I informed the CEO that I hoped he would do whatever he could to get Randal on the team. Luckily for us, we ended up being a better match for him than Microsoft.

Randal had been working as a contractor for years, and was by far the most experienced developer in the company. He knew both front-end and back-end development, but he was hired to the front-end team. I perceived this as somewhat unfortunate, as Randal showed a greater enthusiasm for back-end development and we badly needed someone with experience in the back-end team. Although front-end development is by no means simple and it takes a lot to build a product with a great user experience, it rarely requires the same kind of scientific mindset. At a company in our industry, the back-end is the vulnerable part, and also the part that requires the most from architecture and strategic long-term planning. Randal would happily discuss any back-end issue with me, but I was afraid to ask too often, since what he was measured on was his ability to write front-end code, not his contributions to my team. I brought it up with the CPO, who was still managing the developers at the time, that Randal would be better suited to work with back-end. The CPO pointed out that be that as it may, he had hired Randal as a front-end developer and if he would go shopping for milk, he wouldn't expect to get a haircut instead.

Most of the time when I sought Randal's advice, I already had an opinion about what needed to be done. But that was based on a gut feel-

[13] An *expatriate* is a person residing outside of their native country, and an *expat contract* is usually more beneficial than a local contract as it aims to compensate for the inconvenience of living in a foreign country and culture.

ing, something that I tried not to pay too much attention to. It always fascinated me that when I asked Randal, despite the fact that he was somewhat younger than I, he always had at least one story from a company where he used to work, which had already faced the problem I was asking about. In that way, Randal could always back up my gut feeling with stories from real life, and he was always eager to do so.

It was probably from a sense of despair about the situation at the office and the slow progress we made, that Randal started writing a document describing tasks that no one at the company was currently performing, but had the potential of improving efficiency a lot. The document was very thorough, and explained what could be done in terms of improving internal communication, assessing requirements for resources, identifying what kind of documentation the company needed to provide, developing quality assurance such as code review, implementing strategies for testing code and writing testable code, and developing methods for efficient deployment into production.

To me, it was clear that all of this was something we were in desperate need of, and I believed that we could improve the efficiency several times of the team as a whole, if we could get this right. Randal called the role that he had described *Software Development Manager* and suggested himself as the logical candidate for the job. When I was asked by management what I thought about this I wholeheartedly approved, and a date was set for when Randal would begin his new job.

The date of Randal's change of jobs arrived, and I started using Randal for all kinds of things, from discussing sustainable back-end architecture to having him make a draft design for external documentation. Not until later did I understand that front-end had not yet released him. They still needed him for short-term tasks, and the date when he would eventually be allowed to start his new work was postponed again and again.

My guess is that the state of limbo that Randal ended up in had a huge negative impact on him, and played a large role in his eventual decision to leave the company. You could tell that he wasn't happy about the situation, but he also gradually lost faith in the company's ability to deliver. In the end, a particular event turned out to be the last straw. Randal's Canadian girlfriend had studied linguistics, and was duly requested to verify the English translations of static text strings on the website. She was meticulous in her work, first analyzing the language from a grammatical point of view and then, regardless of what it actually should be, compared with the language used on other related websites. Once Randal had handed over the completed report, he observed how the CPO and some other Frenchmen walked around in the office, read out loud from the report and ridiculed her conclusions. In the end, the CPO stated that he would simply have to throw the report away and make all the decisions himself.

At this point, Randal left the office. It was a Friday morning, and I arrived in the office a little bit later. I noted that Randal wasn't there, but didn't think more about it until he sent me an e-mail message several hours later, stating his intent to hand in his resignation the following Monday but without giving much information about his reasons. I met with Randal and his girlfriend for dinner and drinks, talking things through. While I regretted that Randal would leave the company, I understood his position and encouraged him to follow his instincts. Thus, the following Monday morning Randal went to the office before anyone else, cleaned out his desk, waited for anyone from management to arrive, handed in his resignation and then left the office for the last time.

The CPO brought up Randal's resignation with me the next time we spoke. Without mentioning too much about the reasons behind his decision, he informed me that Randal had resigned, but concluded that it's a loss that we could afford. I wanted to tell him how incorrect that assessment was, and that we had all lost one of our most important and valuable colleagues, but I chose to keep quiet. It's possible that the CPO simply had to display that attitude. After all, the company must find a way to carry on anyway. But I'm almost certain that that wasn't his reason, as he could of course have chosen to say nothing. At any rate, I realized that I could have done nothing to make him understand.

Fortunately, other people had a higher opinion of Randal and the next time Thomas visited, I arranged a lunch meeting between him, Randal and me. Thomas asked several times if there was any way for him to convince Randal to stay. But unfortunately for the company, there wasn't.

4.12 My own shortcomings

My ability to communicate was something that was discussed a lot during my employment. But the opinions were not consistent. Some people kept telling me that I didn't communicate enough. Every time I heard that, I returned to the documentation we had, updated it and posted new versions of it. I sent out even more e-mail messages with what I thought of as thorough descriptions of our achievements so far and our plans ahead. But as some of the people around me also pointed out, very few of the ones requesting more documentation ever read these documents or responded to the messages. While I couldn't tell whether this was because of some political or power game, or if people had genuine problems with the way I communicated, I need to look to my own behavior for answers. As one of my team members described it, my role was often to provide a voice of reason when the requests were vague or impractical. Even though people need to hear it, they don't want to and prefer to "shoot the messenger."

Another possible explanation lies in the compliment that I got a few times, that my e-mail messages were perfect in the sense that they covered all cases and left no stone unturned. This was also my intention, since it was my only defense against ambiguity and uncertainty. By sharing all the assumptions and conclusions, all the situations my solution would work in and all the pitfalls, I could make progress even though I felt that I was missing important pieces of the puzzle. But these messages probably got too long and technical, and by overcommunicating people thought that I didn't communicate enough.

It became obvious during the summer of 2012 that I never got the trust I needed from some on the Product department, something that was much later confirmed as they officially expressed their distrust in me. Even the CEO was upset as the back-end team never seemed to finish its assigned tasks on time. He later told me that the CPO had been furious, as he thought that I had managed my team irresponsibly, several times claiming to be done without actually being done. The CEO also said that Thomas had a similar opinion about this, but when asking him he simply said that there were obvious problems during that summer, but that the cause of these problems couldn't be traced. Personally, I believe that when no one can tell whose fault it is, then the responsibility lies with management rather than with the employees, as management should have better ways to track performance. But at least, I clearly failed at convincing others of how to identify the problems in the organization.

Thomas, once he had become the interim CTO, had an interesting opinion about how I speak with people. Even though I already have the solution, I want others to see it too. But by continuing to talk about it after having claimed to know the answer, I seem unsure. He said that knowing me, if he entered a room where I was in a discussion with someone and he didn't have time to understand the question in detail, he'd always side with me. Yet, I often failed at instilling this sense of trust in others. By asking for the opinion of others, Thomas said that people got the impression that I didn't know what I was speaking about. Then, seeing the results, they would always think "What do you know, he was right! How lucky he is! Again."

During a lunch meeting with the CFO, he informed me about the idea the management team had of me. Their opinion was that "Anders is the smartest guy in town, but we can't wait for three months while he's sketching his plans." I should of course learn how to convey a different image of myself, but in this case I know for a fact that parts of the responsibility must lie elsewhere. In subsection 4.2, I explained my broad background including my participation in the World Finals in programming. If I'm one of the fastest in the world at solving programming problems during a competition, then it doesn't make sense that I

would be one of the slowest at doing the same thing at work. The management team obviously knew about my background, but still didn't question the assumption. Furthermore, subsection 4.8 contains a description of how the CPO asked me to come up with generalized solutions instead of handing me specifications, solutions which I believe that I later delivered according to his request. Arguably, it requires much more time to solve a problem for any unknown future situation, than to do it for a particular purpose with all presently known information available. Asking for generalized solutions implies that time is not the most important concern. Finally, I explained in subsection 4.4 how the CEO with his own eyes could see how I within a couple of hours solved one of our most complex problems, and spent the rest of the weekend with the implementation of it, as soon as I had access to all the specifications. It doesn't even matter if the criticism was partly or even completely correct. Knowledge about my background and the events that took place should have been enough for management to at least consider other possibilities before they pinned the responsibility for all delays on me.

Despite all the criticism, however, it should be mentioned that those who didn't distrust me generally seemed to have high opinions about me. Not long before I left, I was told by someone working in front-end that he thought I was the best manager in the Product and Tech departments. Also, the day Dan found out that I was leaving, he told me that it happened too quickly, and that he had so many questions that he wanted to ask me. He felt like he had missed out on the chance to work together with me and learn from me. It was probably the nicest thing I've ever heard, and I believe that he really meant it.

The criticism wasn't even only about me, but also about the things that I had constructed. The French group disliked the API and rumors about an ugly code base were floating around. Although every aspect of the API had been designed based on our best ideas of how to write simple and structured front-end code, the Frenchmen took it for granted that the API was only the result of workarounds to problems in back-end that we didn't know how to solve, without concern for the needs of front-end. Nevertheless, when my company brought in external consultants to help out on two of our projects, they told the CEO that our API was one of the best ones they had ever seen. Maybe the most remarkable thing isn't what opinions people had about me, but the fact that most people seemed to have a strong opinion. It was as if everything was black-and-white, completely without shades or nuances.

5 Analysis

In this section, the story is analyzed with the help of the theoretical framework which was presented in section 2. The ambition is to show that most of my experiences can be explained through the use of these theories.

The organizational level has been split into two subsections where the first, about organized anarchy, attempts to explain how the organization actively strives to be an organized anarchy while the second one, about the Garbage can model, focuses on how the outcome resembles the one predicted by theory.

The third section deals with scapegoating, which is a group behavior that can at times be observed in either a subgroup of or the whole organization. The final section deals with the individual attribute of narcissism, something that we all have to some extent, but which can have a negative impact on the organization in extreme cases.

5.1 Organized anarchy

One of the most fascinating aspects of the company in question is its continuous struggle to deliver a version of the product that would satisfy the board and the shareholders. Of course, in general no delivery could be considered the final version—a continuous series of improvements is to be expected from most companies in the industry—but it should still be possible to make the shareholders feel that the progress is on track with what has been promised. The maintained status quo of hard work which never completely paid off is interesting partly because the company

was trying to speed up work and move forward but apparently without reaching all the way, and partly because the shareholders seemed dissatisfied but continued to invest even more. These two aspects are likely caused by very different mechanisms, and I intend to analyze the former.

The difficulties that the company was facing were of course not unknown to them. Numerous attempts were made to improve the situation during my employment. Much effort was spent on finding causes and improving structures, but although some progress was made the main recurring themes—late deliveries, lack of specifications, requests to work overtime, inadequate communication and so on—never changed. We even had a management consultant at the office, who tried to change the organization, but without much lasting effect. Thus, I think it's futile to attempt to name a concrete set of steps that are supposed to "cure" the sick parts of the organization. The amount of effort that was wasted trying to analyze the working environment tells me that the problems didn't depend on any specific processes, but on the *paradigm* in which the company operated. By paradigm, I mean a world view from which you get all your goals, ambitions, ideas, tools and processes. Without being able to compare your way of working with that of others, it's nearly impossible to detect that your thoughts are limited by a paradigm, since it constitutes your only frame of reference.

My thought of the company as an organized anarchy is partly supported by the fact that the company regarded itself as a start-up and presented itself in that way (see page 27), since a start-up normally displays similar traits during an early, creative phase.[14]

According to Cohen et al. (1972), an organized anarchy is characterized by three properties—problematic preferences, unclear technology and fluid participation—and the remainder of the subsection will be spent explaining how these all correspond to the behavior of the company.

[14]I've intentionally avoided the use of the word "start-up," since I don't want to warrant the organizational anarchy as a natural phase of the company. Hence, I chose the wordier expression "Young Technology Company" for the title of this thesis.

5.1.1 Problematic preferences

Problematic preferences arise when the preferences are more a collection of ideas than a coherent structure, and the actual preferences are discovered through action. This describes the situation at the company well, given the range of motives and thoughts that were floating around. I will begin by explaining some of the different preferences that existed in the company.

The CEO had a mixture of ideological and financial ambitions. Of course, he wanted his company to have an impact on the Chinese society and to change the way people interact, but he also wanted to make enough money to invest in other ideas. But the way he wanted to achieve this, was by following the latest trends and combining these in ways that no one else had done. This aim made him change his opinion about what the product should be as the trends shifted.

The CPO, on the other hand, wanted to be in total control over the product that we were building. My understanding of his motives is that ideally, he would have had been the sole person responsible for product planning and design, while he was also the head of all of the developers. Even when people came up with ideas, these ideas had to go through him for approval. I believe that he primarily wanted to show off a product that he could truthfully say was completely his achievement. This ambition was probably more important to him than any financial success.

Note that while they both had perceptions of what the product should be, there's no evidence that these were detailed enough to be more than vague ideas.

For my part, as I had early on been told by the CPO that I didn't belong to the product target group and lacked anything beyond a superficial understanding of the business model, I could do very little when I didn't get much specifications. When I did get something, I had to use my imagination and try to cover for any uncertainty by creating versatile solutions. Thus, my aim was to build a technically robust and logical system, an utopia which in itself is not more than a vague idea. In any case, it's not enough as a specification for a product.

The shareholders weren't a very coherent group in themselves. The early ones had been around for a long time, and what they invested in had changed over time. Since then, the company had brought in more investors and informed them about the new course of the company. These people invested in a different set of ideas compared to the original investors, and as the plans subsequently changed again to more resemble the original plans, the early investors tended to welcome the change while the later investors disapproved of them. This tension between shareholder expectations must have been extremely tough to handle, and I can imagine that concessions had to be made in favor of both groups.

The chairman of the board had a deep understanding of the Asian markets, and primarily saw a business opportunity which the company should act upon as quickly as possible.

This is how I perceived the variety of preferences within the company, although these are just a few examples. I felt like everyone had their own opinion about what the product should be. To give further arguments for the existence of problematic preferences, I will attempt to prove a connection between the story and the framework.

The CEO's request that I learned about the competitors' products since we didn't have a specification of our own (see page 22) and his desire that I'd contribute to the purpose of the app (see page 33), the constant pivots, changing directions and changing requirements (see pages 25 and 29) all indicate that the company had problematic preferences. Likewise, the CPO telling me that he couldn't give me detailed instructions (see page 33) and the fact that he even agreed with me that the information I had received wasn't enough for me to build the back-end (see page 31) are basically the same thing, as well as the canceled algorithm lessons and management training (see page 25), the lack of personas and descriptions of user needs (see pages 33 and 34), and finally the book that the company wanted to write but seemed to lack any specifications for (see page 27).

It's of course natural to focus on different things within a company, but when these different functions don't communicate well enough to understand what the common goal is, then the priorities become fuzzy.

5.1.2 Unclear technology

A company has problems with unclear technology if it doesn't understand its own processes, and has to move forward using trial-and-error, pragmatic inventions and experiences of accidents.

There are many examples of how the company had problems with unclear technology. When I first arrived, the agile software development method that was used had obviously been misunderstood (see page 32). When not even the Scrum Master understood the benefits of the method, it's needless to say that none of the developers could learn it either. Even though it was obvious that something was wrong as many of the tasks were never completed, it didn't occur to anyone that the problem was with the method they had adapted. Instead, the developers were blamed for not finishing the tasks they were assigned. I pointed out the problems and a few changes were made, but not nearly enough. As the problem with interrupted Scrum sprints continued to occur, I instead started running Kanban in order to handle this but, again, very few people saw the need for it and that project also ended.

Before I arrived, the API that had been developed had been designed piece by piece by individual developers. Since the basic principles weren't known by these developers (see page 29), the design of a particular feature tended to solve a particular problem at that point in time, but rarely used already existing solutions in a reasonable way, and without providing much room for extensions for future needs. I regard this as a form of unclear technology, since a deeper understanding of how the API should be built to be sustainable would likely have created a better result. It's of course possible that a better way to deal with design is to assign the final say in any decisions to one person or a small team that works together than to distribute this responsibility among all the developers in the company, but that's a separate discussion.

Another example of unclear technology is what

happened when the CDO decided to have his own small team of engineers working directly under him (see page 26). When these engineers needed information to do their job, they would typically come to me to get it. But I was usually under pressure to get things done quickly, and had promised my boss and the CEO to work hard to get something done in a limited amount of time. When this unscheduled request suddenly appeared, without having taken the path through my boss, I could perhaps spend a few hours on the task, but certainly not several days. This caused a conflict between me and the CDO where he accused me of being uncooperative, when in real life it was just a poor understanding of resource allocation and request processes. If the problem was mine (such as if I should have anticipated a certain amount of non-scheduled work and have adjusted for this in my time estimate) or his (as if he should have gone through my boss to get permission to delay my delivery because of his urgent request) I still don't know. It was of course something that I brought up during discussions with my boss, who at the time was the CPO, and we both thought that the deadlines were important and that any requests from the outside should to through the proper hierarchy. But since my team needed to plan weeks in advance and the CDO often had urgent requests that couldn't be scheduled in the same way, no conclusion was reached. On a side note, a better defined company or product goal could have allowed me to make up my own mind about the severity of the requests.

Other examples of unclear technology are when Frank was assigned to help me, but all I got was a load of additional tasks (see page 31) and the interruptions and complaints we had to endure despite working hard (see pages 21 and 29).

5.1.3 Fluid participation

The concept of fluid participation is about how people move in and out of the company, around inside it, between different projects and also how their involvement varies over time. This constant state of fluctuation causes the set of decision makers for a certain kind of choice to change.

The most striking aspect of fluid participation in the company was the explicit goal to improve the team by hiring people who are more skilled than the worst people in the company and then firing the worst ones (see page 26). By definition, this means that the company would never reach a state where it could accept that the work-force is "good enough." I think that the company was expecting that when the right set of people was on the team, the ideas that would lead to success would appear, like a chain-reaction can start by itself in a nuclear reactor when the conditions are just right.

A consequence of hiring western people for a Beijing-based company was that the company needed a certain amount of flexibility in terms of where people were working. Although it was all for good reasons, it still meant that the chairman of the board had to fly in from Taiwan once in a while, that the CFO lived, until the spring of 2013, in Shanghai and that the interim CTO traveled from Texas to Beijing for a maximum of one week per month. It should be pointed out that there's not necessarily anything wrong in having a staff that travels a lot. Nevertheless, it helps us to classify the company and to better understand its processes.

The CEO clearly wanted to promote fluid participation, since he asked me to work on issues with mobile communication even though it would have been someone else's responsibility anywhere else, and told me that he wanted a flat organization (see page 28). The fact that I was moved to work on the iOS project (see page 22) shows that the company believed that the combination of participants could hold the key to solving a problem.

It can be theorized that the recruitment bonus (see page 26) that promoted hiring people on a short term, or even hiring bad people to have them replaced, was a way to increase the fluidity so that there would be more opportunities for participants to match solutions with problems, but there's not enough evidence of such an intention.

5.2 Garbage can model

The Garbage can model predicts a behavior where ideas are tried out and then discarded, where un-

clear goals allow the direction to change completely, and where every employee can help to define the target. Indeed, these are behaviors that I could observe during my entire employment. Although the company strove for a flat organization and saw the generation of ideas from employees as a central part of this, it should be noted that the Garbage can model slows down progress.

The willingness of the company to quickly change directions and to do "pivots" (see page 25), already covered in subsubsection 5.1.1, shows that the company lacked a detailed plan. Still, problematic preferences don't explain *why* the company would end up doing these quick changes, only that they are allowed to happen. The Garbage can model helps to explain it because of the way it goes about solving problems. Any solution is only developed for a while, until something else comes along that looks better. At that point, the older solution is thrown into the garbage can. The company is particularly steered towards creativity instead of productivity as it at least seemingly favors the one who comes up with ambitious ideas—however impractical or far-fetched—while it often disdains and antagonizes the one who wants to discuss and asks for specifications in an attempt to implement these ideas.

On a smaller scale, the Garbage can model explains why it was difficult to make Scrum work. The cycle of finding new solutions, trying them out and then rejecting them was simply too quick. That explains why we noticed an improvement when changing to Kanban (see page 20).

The reason for why I never got the chance to work on my thesis (see page 20) might be that the influx of new solutions to try out was too great, making it impossible for management to plan ahead. The same thing goes for promises about vacation to people who had been working hard (see page 39).

On a side note, I was trying to create more structure within the back-end team, but have to admit that I didn't completely succeed. The e-mail messages that I sent out, which were never answered (see page 30), can in a way be said to have been thrown into the garbage can, although not intentionally. Many of the suggestions hadn't been written down anywhere but in the message, and when the message wasn't answered, the suggestion got lost in my mailbox while I focused on other solutions. Another kind of garbage that we had problems dealing with was uncompleted code that was left in our code base, since no decision was made to remove the old solutions (see page 32).

5.3 Scapegoating

I believe that what seemed to be a constant search, that primarily the French group was occupied with, for an individual in the organization responsible for all the problems we had been experiencing, is a textbook example of corporate scapegoating. It is unclear to me, however, what kind of consequences an individual should on average expect when being singled out by such a group, other than the social rejection. Some of the people who were blamed were indeed let go (see pages 25 and 26) but others, including me, could stay. In other words, I don't know how much of a driving force this activity constituted when measuring people's ability to perform. Still, I am certain that this kind of "mild" scapegoating helped to maintain the internal reputation within the group. This also suits well with the fact that I often heard them speak highly of the skills of other people in the group, but rarely about anyone else (see page 28).

What would interest me to know is how aware one needs to be of this process to participate in it. Wilson (1993) never explains whether an intentional agent needs to be conscious. I believe that this kind of activity can be performed through a sense of necessity, without anyone realizing what the underlying motivations really are. In that case, the organization will really believe that the scapegoat is guilty.

I think that the discussion about whether the scapegoat is completely innocent, or to some extent guilty, is irrelevant. The problem with blaming a scapegoat is primarily that the real issue never gets fixed. For example, the issue might be a flaw in how requirements are collected, a boss that should be replaced or a general need

for training that should be addressed. By firing someone, and assigning the blame for previous failures to that person, management chooses the path of least resistance. The firing might even be seen as a "tough decision," a decision that ensures that the company will do better in the future, and this promise could in itself be sufficient to attract another capital injection from the investors. But at this point, the matter is considered solved by everyone, and given that it's solved no further improvements are needed. Unfortunately, this doesn't make the issue any less likely to appear again in the future.

Looking at my own behavior, I notice that I have the same tendency to search for scapegoats. After the dysfunctional member of my team had been fired (see page 36) and everyone started to notice improvements, I reached the conclusion that he was responsible for the problems that we had been facing. In reality, this wasn't completely true. Sure, my team had severe problems during the entire time he was there and, sure, I had my concerns about him all the time. But the problem wasn't with him. There were other options available at earlier stages, for example reassignment, tougher requirements or longer talks. I had felt that my hands were tied, as I had been asked by my superiors to keep him in my team and see to it that his opinions would change, in order to motivate his friends in the company. That was an inappropriate request that couldn't be fulfilled, and which had consequences for both the morale in the team and our results. The blame should be directed either at my superiors for asking me to sacrifice my team to make others happy, or at me for not strongly objecting to such a preposterous request. But it was wrong of me to think that a team member was responsible for demotivating my team for months. There was definitely enough time there to react.

Ordoñez (2009) wrote about a reluctance to hire experts during bad times. Given that my company had been experiencing problems delivering a product for a long time, this theory might shed some light on why I thought that some of the most capable people at the company were at times the most oppressed, since admitting that an employee is an expert in their respective field means that it's more difficult to later take it back. It could also explain why the CPO pointed out several times that he was never consulted about whether I should be hired. The same goes for how the CPO could tell me that the loss of Randal, one of the most experienced people that I've ever worked with, was something that we could bear (see page 40). In bad times, non-experts are thus simply more valuable than experts. And with enough charisma, you'll of course even have the possibility to decide who's an expert and who isn't.

The same theory also raises questions about how the CDO could get a bonus for every person he recruited (see page 26), given that there will already be a bias for hiring non-experts that could later be fired. The more people that need to be replaced, the more bonus will be paid out for the recruitment. One way of looking at it would be that it actually creates an incentive to find unskilled people who could act as scapegoats. But regardless of whether there is such a motive, this must at least increase the tendency to look for people who are not top-class.

Did the Swedish group take part in scapegoating in the same way as the French group? To be fair, the French were often challenged by the Swedes, but it's uncertain whether this behavior, except in a few cases (see page 22), can be attributed to scapegoating. Our criticism often concerned their unwillingness to discuss technical matters, the fact that they almost always left the office at the same time every evening despite the pressure the department was under and that other people worked late, and their sometimes disrespectful way of speaking, which in some cases made people lose their motivation to work. While these relations were of course bad in themselves and should be solved, it is doubtful if they met the conditions that constitute scapegoating. The events were well-documented, and the criticism was generally not aimed at a particular individual but at the behavior of the group as a whole. Their own way of explaining our attitude towards them as racism also tells me that they themselves didn't think of this as aimed at any particular individual.

For my part, the first time I became the scapegoat can probably be attributed to guilt by association, resulting in my exclusion from the group (see page 35). Later on, I could be blamed for the department's inability to communicate (see page 41) or asking too many questions (see page 32). It didn't matter that I worked longer hours than any of those who complained (see pages 21, 23 and 24). I also noticed how my work could be judged without anyone actually looking at it (see pages 36 and 37), and I could get the blame for delays with the assumption that I'm unable to work quickly (see page 41). But other people in power could also get the role as scapegoat, even the CEO (see page 35).

A theory I have is that the prevalence of scapegoating at the company made people more defensive. For example, a reason for not making any decisions could be a fear of failing (see page 34).

5.4 Narcissistic leadership

According to the theories about destructive narcissistic leaders, some companies are more likely to attract such leaders than others. My company completely lacked a system for performance reviews. Also, the perspective was almost always short-term, which makes performance measurement difficult.

One interpretation of the French group is that it formed around a leader in the way described by Lubit (2002), since the group members spoke highly of each other, could easily devalue people outside of the group and refused to focus on details.

For the record, everyone displays some level of narcissism, but I believe that at my company, some of the cases were noteworthy. Specifically, my belief is that the CPO displayed tendencies resembling destructive narcissism. Among the signs were his friends that surrounded him and the scapegoating that they were engaged in (see page 28), the desire to keep all the responsibilities to himself and not work in a design team, the polarized opinions of him when comparing his subordinates with his peers and superiors, and the attempts to disparage some of his subordinates while having troubles accepting when people tried to respond in kind. He didn't care for the work environment of his subordinates (see pages 26 and 27) and their desires (see pages 35 and 38). He didn't care about Randal (see pages 39 and 40) and wanted me to be grateful to him for simply providing me with the tools to do my job (see page 38), but might later have perceived me as a threat and wanted to reduce my influence (see pages 21 and 37). He misunderstood and ridiculed other people's work (see pages 37 and 40). Finally, although I can't know if it's because of him or someone else, people seemed to think that his skills were a superpower (see page 29) and the French group tended to see everything in black-and-white (see page 42).

I made the mistake of questioning the way the CPO worked, something which caused me to become even more alienated. Lubit (2002) tells me to appear to admire the narcissist, while finding other ways of solving the problem. Interestingly, however, I instinctively started to document everything and requesting specifications without knowing that this is what Lubit recommends, but it turned out that the Garbage can way of creating solutions without knowing the problems got in the way, and I was criticized for not being able to do my job without clear instructions. Furthermore, once I was reluctantly handed some bits and pieces of information, I could then be criticized for not being able to do my job even though I had received my instructions. In other words, a destructive narcissist seems to thrive in an organized anarchy, as the performance indicators can always be adjusted in retrospect.

6 Conclusions

This thesis has attempted to describe an organization using theories of organized anarchy and the Garbage can model, and how they allow mechanisms such as scapegoating and the rise of destructive narcissistic leaders.

The company has actively promoted individual ideas about where the product should be going, or in other words problematic preferences. The company has unclear technology, where it

in some cases (such as Scrum) wanted to implement the technology but misinterpreted it, and in other cases (such as the API) not even required that people who were assigned to work with it agreed to learn it. Furthermore, the company has fluid participation because people are traveling back and forth, moving between teams and because of the high staff turnover. Hence, it meets the requirements to be called an organized anarchy. The production can be described according to the Garbage can model, as many of the company's solutions never reach the point of release, and as development can't even carry on for a few weeks without being interrupted with new ideas that require the tasks to be reprioritized.

There are strong indications that one or more destructive narcissists are operating in the company, as suggestions can be openly mocked, talents disregarded and neglected, and people unwilling to care about details, working in teams and so on.

The presence of scapegoating at the company was obvious. Several people were blamed for problems during my time there, and some of them were subsequently fired. Scapegoating can be the result of narcissistic leadership, but it can also occur on its own.

Regardless of the root cause, the four systems support each other. Organized anarchy and the Garbage can model are intertwined. They allow scapegoating and destructive narcissism to exist, since they don't provide control functions to detect such behavior. Scapegoating attracts the attention when an organization is looking to cast blame, giving an excuse for other behavioral patterns to keep going. Destructive narcissism, finally, causes scapegoating in order to remain impeccable, and will attempt to get rid of anyone who constitutes a threat to the power.

To answer the research question about how theories of culture and teamwork can explain irregularities in productivity within a young company with a largely inexperienced staff, I will give an answer in three points:

1. A company which promotes creative and entrepreneurial values, such as distributed responsibility, trial-and-error based technology and high mobility between tasks, stands the risk of creating solutions to problems that no one has asked for. While it makes it easier to find new business opportunities, productivity will likely suffer.

2. If people are reluctant to search within themselves for causes of problems, the company might have to suffer from scapegoating. If scapegoating is used to solve problems with investors, for example, then a bias might develop for avoiding recruiting highly skilled workers as it's easier to blame problems on a low-skilled worker. This affects productivity in the long run.

3. If the company doesn't conduct thorough evaluations and rather recruits people based mostly on recommendations and interviews, then there's a risk for attracting narcissistic leaders. Such leaders are normally not as productive as others, and they take credit for other people's work and attempt to get rid of talented people, who might be considered a threat.

It's reasonable to believe that all of these situations are likely to occur with an inexperienced staff, which doesn't recognize the symptoms.

The patterns of causality are complex, and events and processes are interconnected. Still, it's important to start the search for root causes in the basic ideology and policies of the organization, and not the employees. With the exception of a few people that I got along with really well, I agreed on some things with each colleague, and disagreed on others. No one was completely right or completely wrong, and no one was completely good or completely bad. People's ambitions and abilities are shaped by their surroundings. By looking for responsible individuals, we're just starting the scapegoating cycle all over again. Of course, an employee who isn't cut out for the job might in the end have to be transferred—there's no way to completely avoid that—but the organization should also try to understand what went wrong to make this necessary. After all, if for example a malfunctioning manager were to be removed a vacuum would

be left behind, allowing for someone else to rise. With nothing changed except for the removal of one person, what reason is there to believe that the new leader will be any better than the predecessor?

7 Discussion

In the final section of the thesis, I will give my view of what difference the thesis has made for me, and what difference I believe it can make for researchers and business people.

7.1 Reflection

My own role in the story is both as an observer and as a participant. A possible interpretation of my observations is that I'm regarding myself as an impartial researcher and describing only the faults in others. However, this is not my intention, and I've been forced to look at my own behavior as well in the search for answers. Beyond a need for me to express my opinions earlier and more clearly, I not only participated in but also agreed with the attempts within the Product department to isolate itself from external micromanagement, and the scapegoating that took place after the French lead mobile developer was fired.

It has been important for me to write this story down. During the first year in China, I spent nearly every day at the office. Except for a ski trip I didn't travel at all in China, and I hardly even saw anything of Beijing. When I finally had to resign, the only thing I could bring with me was what I had learned and experienced. Because of that, I wanted this thesis to tell both the story about me, and about the company itself.

People asked me if I was happy when certain people that I had been in conflicts with left the company. Strangely enough, I wasn't. Maybe it was because I had already filled my diary with every emotion that I had felt over the last months. I just knew that I wanted to be treated fairly, but I didn't want it to be at the expense of someone else. When writing this down, I want everything to be as accurate as possible, and I don't hold a grudge against anyone.

I've tried to be honest about my own mistakes as well as those of others, and I've also tried to find theories that I really believe can make a difference in understanding what was going on. My feeling is that I've succeeded.

My concerns are that I couldn't get feedback from some of the people who are the most important in this story. It's risky to claim to know someone who doesn't even speak to you. Also, I'm not completely satisfied with the structure I chose to tell the story, but I'm not sure I could have done any better. I created themes for events that had something in common, but the themes could probably have been limited in other ways.

In addition, I wish that I could have gotten further with my studies about ethics. The theories I present are contradictory.

I would recommend others who are choosing a similar research method to use a technique similar to mine, with daily diary entries. Write about everything. Whom did you have lunch with? What movie did you watch in the evening? This is information that will help you remember the day more clearly. When writing, don't focus too much on structure at first. Just get all your thoughts down. Think about it to see if you got all your thoughts out of your head, otherwise continue to write. Let the text rest for a bit, then read it and check if you need to rewrite it. Be prepared that you might not like the first version.

7.2 Implications

A valid point the reader can make after having read through this thesis, is that the analyses and conclusions are based solely on my own subjective thoughts and lack the generality to explain the phenomenon even in the rest of the company, let alone anywhere else in society. However, it surprised even me to notice how well I could predict the development in the company in the months following my resignation, using only the framework I've described. My former colleagues also helped me to verify the significance. While they all maintained that the story I told is my own, no one disputed the relevance of the framework when attempting to figure out how the company

is operating. On the contrary, those who read the thesis usually recommended it to someone else, who in turn came to me and asked if they could have a copy. As for the usefulness for others, I can only say that in the short time which has passed since I finished writing, I've discussed the topic with several friends outside of the company, who had remarkably similar stories to tell. They, too, asked to get to read the thesis. While it remains to be seen if the conclusions are general enough, at least it seems as if my story isn't an isolated incident.

Researchers might be interested in this thesis as it's an attempt to use analytic autoethnography to describe a business environment. The ethical considerations that I've touched upon can certainly be developed further.

Managers and other business people might instead be interested in the idea that an organization can be unproductive despite using the best technology and hiring the most clever people. But it's not obvious that an organized anarchy will be harmful. Indeed, such an organization is usually very creative but, when combined with a pressure to deliver as the projects are already overdue and about to run out of money, certain mechanisms can be set in motion to deal with the pressure, since an organized anarchy typically doesn't have a control function to identify these unhealthy processes. At the very least, by learning about these theories managers can understand their organizations better and maybe find a way to make them more productive by first knowing how they work. A good place to start when evaluating the health of the organization might be to implement 360-degree feedback, and possibly continuing by increasing the amount of structure in the organized anarchy to improve accountability. Remember that different people can hold different parts of the answer. A manager shouldn't expect to always have all the information.

References

Anderson, Leon (Aug. 2006a). "Analytic Autoethnography". In: *Journal of Contemporary Ethnography* 35.4, pp. 373–395 (cit. on pp. 3, 6, 7).

– (Aug. 2006b). "On Apples, Oranges, and Autopsies". In: *Journal of Contemporary Ethnography* 35.4, pp. 450–465 (cit. on p. 7).

Berger, Peter L. and Thomas Luckmann (1966). *The Social Construction of Reality. A Treatise in the Sociology of Knowledge*. Garden City, NY: Anchor Books. 240 pp. (cit. on p. 5).

Cohen, Michael D. and James G. March (1974). *Leadership and Ambiguity: The American College President. A General Report Prepared for the Carnegie Commission on Higher Education*. New York: McGraw-Hill Book Company. 270 pp. (cit. on p. 11).

Cohen, Michael D., James G. March, and Johan P. Olsen (Mar. 1972). "A Garbage Can Model of Organizational Choice". In: *Administrative Science Quarterly* 17.1, pp. 1–25 (cit. on pp. 10, 11, 43).

Congress of Qualitative Inquiry (May 2007). *Position Statement on Qualitative Research and IRBs*. Discussion draft. URL: `http://quig.psu.edu/docs/IRB_PositionStatement.pdf` (cit. on p. 9).

DiBella, Anthony J. (1992). "Planned Change in an Organized Anarchy: Support for a Postmodernist Perspective". In: *Journal of Organizational Change Management* 5.3, pp. 55–65 (cit. on p. 11).

Ellingson, Laura L. and Carolyn Ellis (2008). "Autoethnography as Constructionist Project". In: *Handbook of Constructionist Research*. Ed. by James A. Holstein and Jaber F. Gubrium. New York: The Guilford Press. Chap. 23, pp. 445–465 (cit. on pp. 5, 7).

Ellis, Carolyn (Nov. 1993). "'There Are Survivors': Telling a Story of Sudden Death". In: *The Sociological Quarterly* 34.4, pp. 711–730 (cit. on pp. 5, 9).

– (2004). *The Ethnographic I. A Methodological Novel about Autoethnography*. Walnut Creek, CA: AltaMira Press. 448 pp. (cit. on p. 5).

Ellis, Carolyn S. and Arthur P. Bouchner (Aug. 2006). "Analyzing Analytic Autoethnography. An Autopsy". In: *Journal of Contemporary Ethnography* 35.4, pp. 429–449 (cit. on pp. 6, 7).

Gemmill, Gary (Nov. 1989). "The Dynamics of Scapegoating in Small Groups". In: *Small Group Research* 20.4, pp. 406–418 (cit. on p. 12).

Hayes, Sherman L. and Patricia B. McGee (Mar. 1998). "'Garbage can decision making' in a 'structured anarchy' for your CWIS. Could you translate that for me please?" In: *Campus-Wide Information Systems* 15.1, pp. 29–33 (cit. on pp. 10, 11).

Kets de Vries, Manfred F. R. and Danny Miller (June 1985). "Narcissism and Leadership: An Object Relations Perspective". In: *Human Relations* 38.6, pp. 583–601 (cit. on pp. 13–15).

Lubit, Roy (Feb. 2002). "The Long-Term Organizational Impact of Destructively Narcissistic Managers". In: *The Academy of Management Executive* 16.1, pp. 127–138 (cit. on pp. 13–15, 48).

Ordoñez, Guillermo L. (July 2009). *Reputation from Nested Activities: The Inefficient Effects of Scapegoating*. Research Department Staff Report 430. Federal Reserve Bank of Minneapolis (cit. on pp. 13, 47).

Oxford, Cliff (Sept. 2012a). "What Do You Do With the Brilliant Jerk?" In: *The New York Times. You're the Boss Blog* (26). URL: http://boss.blogs.nytimes.com/2012/09/26/what-do-you-do-with-the-brilliant-jerk/ (cit. on p. 15).

– (Oct. 2012b). "Further Thoughts on the Brilliant Jerk". In: *The New York Times. You're the Boss Blog* (3). URL: http://boss.blogs.nytimes.com/2012/10/03/further-thoughts-on-the-brilliant-jerk/ (cit. on p. 15).

Reed-Danahay, Deborah E. (1997). "Introduction". In: *Auto/Ethnography. Rewriting the Self and the Social*. Ed. by Deborah E. Reed-Danahay. Oxford: Berg Publishers, pp. 1–17 (cit. on p. 7).

Runge, Sarah (Oct. 2009). *The Art of Scapegoating in IT Projects*. PM Hut. URL: http://www.pmhut.com/the-art-of-scapegoating-in-it-projects (cit. on pp. 12, 13).

Safire, William (Apr. 2007). "Fall Guy. Taking wraps off taking raps". In: *The New York Times Magazine* (29). URL: http://www.nytimes.com/2007/04/29/magazine/29wwlnsafire.t.html (cit. on p. 13).

Spicker, Paul (2007). "Research without consent". In: *Social Research Update* 51. URL: http://sru.soc.surrey.ac.uk/SRU51.pdf (cit. on p. 9).

Tolich, Martin (Dec. 2010). "A Critique of Current Practice: Ten Foundational Guidelines for Autoethnographers". In: *Qualitative Health Research* 20.10, pp. 1599–1610 (cit. on p. 9).

Wilson, P. Eddy (Oct. 1993). "The Fiction of Corporate Scapegoating". In: *Journal of Business Ethics* 12.10, pp. 779–784 (cit. on pp. 12, 46).

Index