STOCKHOLM SCHOOL OF ECONOMICS
DEPARTMENT OF MANAGEMENT
MASTER THESIS
MAY 2016

Controlling motivation of software engineers in Swedish technology start-ups

A multi-case study

Axel Elner (50157) & Tobias Kanevret Lööf (22253)

Abstract

Sweden has in recent years started to profile itself as a technology hub due to its rapid increase in technology (tech) start-ups. Though management control systems (MCS) and motivation have separately been shown to increase the growth of start-ups and performance of software engineers (SE), the literature is still in a surprisingly nascent state when combining the two in a start-up setting. This paper aims at addressing this gap by specifically looking at how MCS as a package are designed in tech start-ups to act as an external factor that affects the self-determination of individual SEs, and thus boost their performance. To answer this the study first looked at what motivators affects the self-determination (autonomy, competence, relatedness) of an individual SE and secondly how the design of MCS as a package can enable this self-determination. Using an explorative, multiple case study and collecting data using qualitative in-depth interviews in four tech start-ups we first find specific motivators of SE related to their self-determination and secondly that there is an interrelatedness between motivation and MCS as a package. Specifically, autonomy, competence and relatedness are satisfied by specific configurations of both formal and informal control systems that together can have a positive effect on SE motivation. Moreover, the use of administrative controls such as retrospectives allows SEs to affect the MCS package and adapt the design to the SEs basic psychological needs.

Keywords: Tech start-ups, management control systems, motivation, MCS as a package, basic psychological needs

Supervisor: Mikael Samuelsson

Examiner: Karin Felner

Acknowledgements

We could not have completed this thesis if it were not for the generous help of the four start-ups that we have been interviewing. We would therefore like to thank all the managers and software engineers participating in the study for their hospitality and interest in our work. We are also very thankful to the persons participating in our pre-study as they have led us into this subject and guided our initial research. We would also like to provide a special thanks to our great supervisor Mikael Samuelsson for his invaluable feedback and to Martin Carlsson-Wall for his great advice.

Lastly, we would like to send our gratitude to the faculty at the department of management for two great years of master studies. It's been a true pleasure!

Table of content

1. Introduction **5**
- **1.1. Problem formulation** **6**
- **1.2. Purpose and research question** **6**
- **1.3. Expected contribution** **7**
- **1.4. Scope of study and limitations** **7**
- **1.5. Method and thesis roadmap** **8**

2. Pre-study **9**
- **2.1. The need for a pre-study** **9**
- **2.2. Pre-study design** **9**
- **2.3. Findings** **10**
- **2.4. Conclusions from the pre-study** **10**

3. Literature Review and Theoretical Framework **11**
- **3.1. Management control systems: a review** **11**
  - 3.1.1. MCS as a package 12
  - 3.1.2. Management control systems in early-stage companies 13
- **3.2. Defining motivation theories** **14**
  - 3.2.1. Individual motivation in organizations 16
- **3.3. Identifying research gap** **18**
- **3.4. Theoretical framework** **19**
  - 3.4.1. Management control systems as a package 19
  - 3.4.2. Motivators for software engineers 20
  - 3.4.3. Fulfilling needs to fulfil goals 21
- **3.5. Forming our theoretical framework** **22**

4. Methodology of Main Study **24**
- **4.1. Methodological fit** **24**
- **4.2. Research approach** **24**
- **4.3. Case study** **25**
  - 4.3.1. A multi-case study 25
  - 4.3.2. Selection of companies 26
- **4.4. Data collection and documentation** **27**
  - 4.4.1. Choice of people 27
  - 4.4.2. Design and documentation of interviews 28
- **4.5. Analysis and interpretation of data** **29**
- **4.6 Data quality** **29**
  - 4.6.1 Reliability 30
  - 4.6.2 Validity 30
  - 4.6.3 Transferability 30

5. Empirics **31**
- **5.1. Company A** **31**
  - 5.1.1. Administrative controls 31
  - 5.1.2. Planning controls 32
  - 5.1.3. Cybernetic controls 32
  - 5.1.4. Reward and compensation controls 33
  - 5.1.5. Cultural controls 34
  - 5.1.6. Motivation of software engineers 34
- **5.2. Company B** **35**
  - 5.2.1. Administrative controls 35
  - 5.2.2. Planning controls 36
  - 5.2.3. Cybernetic controls 37
  - 5.2.4. Reward and compensation controls 37

5.2.5. Cultural controls ... 38
5.2.6. Motivation of software engineers ... 38
5.3. Company C ... 39
5.3.1. Administrative controls ... 39
5.3.2. Planning controls ... 40
5.3.3. Cybernetic controls ... 41
5.3.4. Reward and compensation controls ... 41
5.3.5. Cultural controls ... 42
5.3.6. Motivation of software engineers ... 42
5.4. Company D ... 43
5.4.1. Administrative controls ... 43
5.4.2. Planning controls ... 44
5.4.3. Cybernetic controls ... 45
5.4.4. Reward and compensation controls ... 46
5.4.5. Cultural controls ... 46
5.4.6. Motivation of software engineers ... 46
6. Analysis ... 48
6.1. Motivation of software engineers ... 48
6.1.1. Enabling autonomy through proper task definitions, constraints and influence ... 48
6.1.2. Satisfying competence with personal development and complex new work ... 50
6.1.3. Creating relatedness through team and recognition ... 52
6.2. MCS package configurations affecting fulfilment of psychological needs ... 53
6.2.1. Autonomy satisfied through specific MCS package ... 53
6.2.2. MCS package enabling satisfaction of competence ... 57
6.2.3. Relatedness satisfied through a combination of controls ... 60
7. Conclusions ... 62
7.1. What motivates software engineers ... 62
7.2. How MCS packages affect motivation of software engineers ... 64
8. Discussion of contributions, limitations and further research ... 66
8.1. Main contributions ... 66
8.2. Practical implications ... 67
8.3. Limitations ... 67
8.4. Areas of further research ... 67
References ... 68
Printed sources ... 68
Other Sources ... 73
Appendices ... 74
Appendix 1. Pre-study interviews ... 74
Appendix 2. Different types of control and its components of management control systems ... 75
Appendix 3. Interview sample for the main study ... 76
Appendix 4. Questions posed in the Pre-study ... 76
Appendix 5. Interview questions posed in the main study ... 77
Appendix 6. Summary of MCS package & motivation in case companies ... 81
Appendix 7. MCS package satisfying the three basic psychological needs ... 83

1. Introduction

Entrepreneurship in Sweden has almost doubled since 2007 (GEM 2014), were the IT sector has grown at a pace 10.3 % annually and resulted in over 22,000 tech businesses in the city of Stockholm alone (Knowledge at Wharton, 2015). This growth has seen Sweden propel into one of the most prolific tech hubs in the world on a per capita basis, producing 6.3 billion-dollar companies per million people, according to a recent report from the investment firm Atomico (The Telegraph, 2015).

High growth targets generally characterize tech start-ups, and to achieve this management control systems (MCS) has proved an important factor to consider as it has proved to enable higher growth rates of employees, revenues and valuation (Davila & Foster, 2005; 2007; 2010). MCS support managers in their attempts to align the behaviour of individuals to organizational (strategic) goals (Otley & Berry, 1980; Simons, 1995; Langfield-Smith, 1997; Chenhall, 2003). This wide description involves several different frameworks from management literature and has resulted in different conceptualizations of MCS (Malmi, 2013). Recently, a concept that aims at bringing different types systems and concepts (such as financial measure and planning) together is MCS as a package (Malmi & Brown, 2008; Strauss et al., 2012). This provides an advantage as the concept takes a holistic perspective to view how well aligned different systems are to adjust the goals of individuals with that of the organization (Malmi, 2013).

Meanwhile, motivation in IT companies is of relevance as it is increasingly cited as a particularly pernicious people problem, and motivation of software engineers (SE) are essential for individual performance, and to either succeed or fail in deliver good-quality software products (DeMarco and Lister's, 1999; Roberts et al., 2004; Beecham et al., 2008). Motivation is referred to as "… means to be moved to do something" (Deci & Ryan, 2000) and to become motivated humans need to fulfil three basic psychological needs: autonomy, competence and relatedness. Though, there has been limited research that directly relates MCS with the basic needs, external factors within the work environment have been shown to affect motivation (Bard et al. 2004; Deci & Gagne, 2005; Chen et al. 2014). Therefore, the design of the MCS package is expected to affect the motivation of employees and thus their performance, which is a gap that this paper aims at investigating further.

1.1. Problem formulation

Most prior research has focused on specific formal control systems, such as financial based control, within large, established companies (Otley & Merchant, 2007; Davila & Foster, 2005; 2007). This provides challenges as start-ups both rely on formal and informal control systems (Collier, 2005; Hutzschenreuter, 2009). Additionally Malmi & Brown (2008) suggest that one should consider different control systems of MCS simultaneously as it enables to see how well-aligned different systems are to direct employees goals with that of the organization (Malmi, 2013). The research of MCS as a package is however still in a nascent state and the framework has mostly been tested on a conceptual level in a start-up context (Haustein et al., 2014).

Moving on, though motivation has proved to affect performance of individuals and performance rankings of SE (Bard et al. 2004; Roberts et al. 2004), factors that enable motivation from the perspective of an individual SE is thus far limited (Beecham et al., 2008). Together, this implies that though MCS and motivation is crucial for technology (tech) start-ups, there is little existing research looking at motivation through the lens of management control in tech start-ups. Specifically, the literature has thus far not viewed the dynamics of MCS as a package and its relationship to individual's self-determination (degree of fulfilment of the three basic psychological needs) and thus individual performance.

1.2. Purpose and research question

This paper aims at giving further theoretical insights into the area of management by studying MCS and its relationship to motivation of SE by studying four Swedish VC backed tech start-ups. As a consequence of the pre-study (section 2) and literature gap (section 3.3) this study will look at how MCSs as a package are designed to affect self-determination of SE and thus their performance. In effect, this paper will investigate the following research question:

How does the design of the MCS package affect the motivation (self-determination) of software engineers in the context of Swedish tech start-ups?

To answer the research question and research gap that is presented in Figure 1, it is of importance to understand (1) how SE are motivated in tech start-ups, and (2) how MCS as a package affect motivation of SE.

To summarize, these two themes will lay the foundation of this study:

- What motivates SE in tech start-ups
- How does MCS as a package affect motivational factors of SE

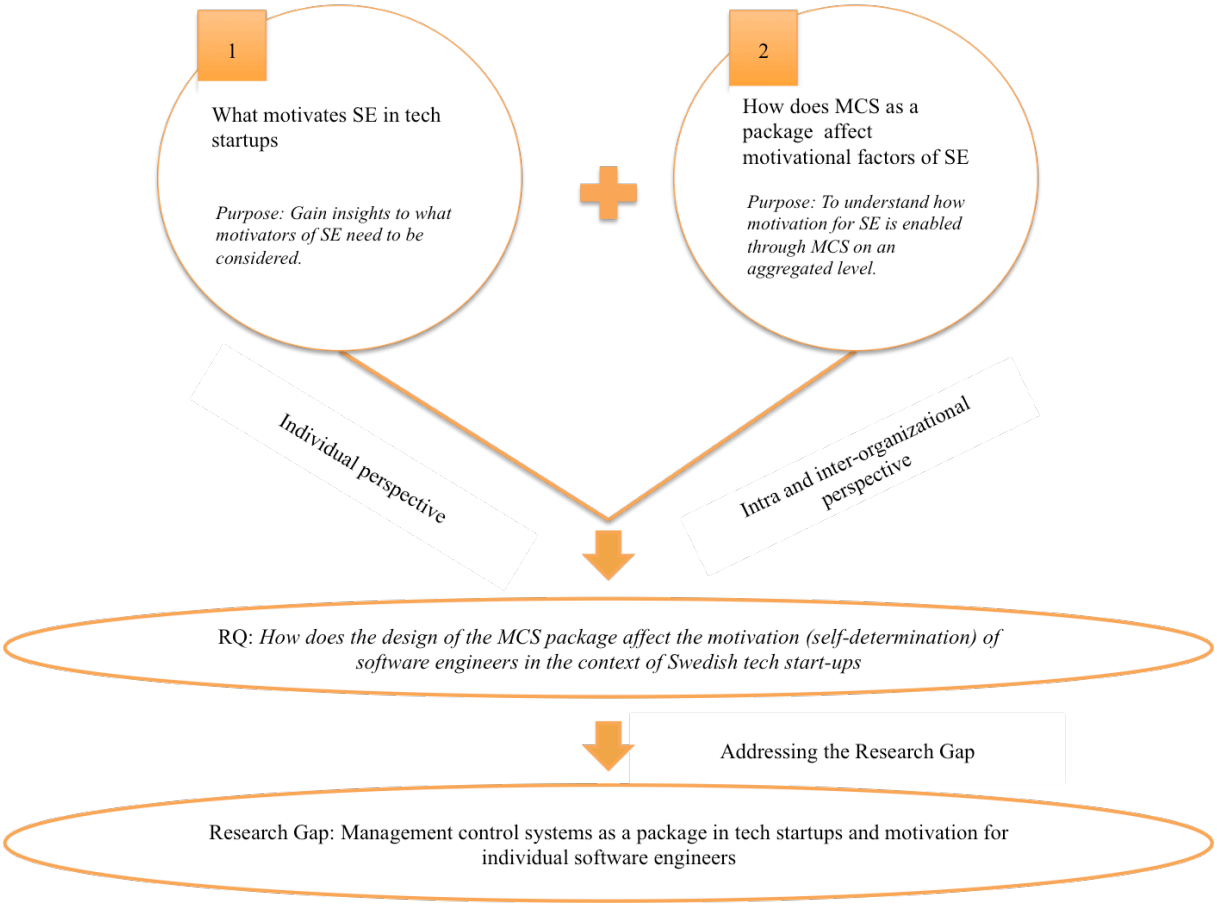

Figure 1. The thesis research gap and the plan of how to address it.

1.3. Expected contribution

This thesis aims to contribute to the field of management studies by providing managers in tech start-ups and researchers within the field knowledge of how the design of the MCS affect the motivation of software engineers, by complementing the existing MCS theory. Additionally, we also aim at presenting empirical contributions to (potential) entrepreneurs in how they should work with MCS, as well as increasing the awareness of MCS and its potential usefulness, which, as for now, is often seen as something negative in start-ups (Davila & Foster, 2010).

1.4. Scope of study and limitations

The scope of this paper has been thoroughly narrowed to provide more insight into the subject area. We focus on companies that have received funding from Swedish venture capital firms, as this increases the likelihood of having implemented formal MCS (Davila & Foster, 2010).

Also, the case companies have their main office in/close to Stockholm, Sweden, as there are proximity advantages in this when conducting the interviews.

As part of the limitations the research design takes place over a limited time period. This is reasonable considering the time frame of the research paper. Further on, to provide a more coherent framework the study will be conducted using a multi-case study. It does not, however, provide a true holistic overview of each and entire company as there was no possibility of interviewing every person and/or department within the company. Also, even though all companies are technology companies, some do differ in terms of their business model, technical platform and age of the company, which can affect the study's results.

1.5. Method and thesis roadmap

This paper is structured in eight parts. After the introduction, a pre-study will be conducted that digs deeper in how MCS are used in tech start-ups and how employees are motivated. After viewing how the use of MCS looks like in tech start-ups and motivation is enabled, the scope will be narrowed and theory will be built upon this. During the methodology section the use of a multi-case, explorative study is motivated, as well as the use of an abductive approach. Following that are empirical findings and after having made an analysis, the paper will conclude its findings and end with a discussion.

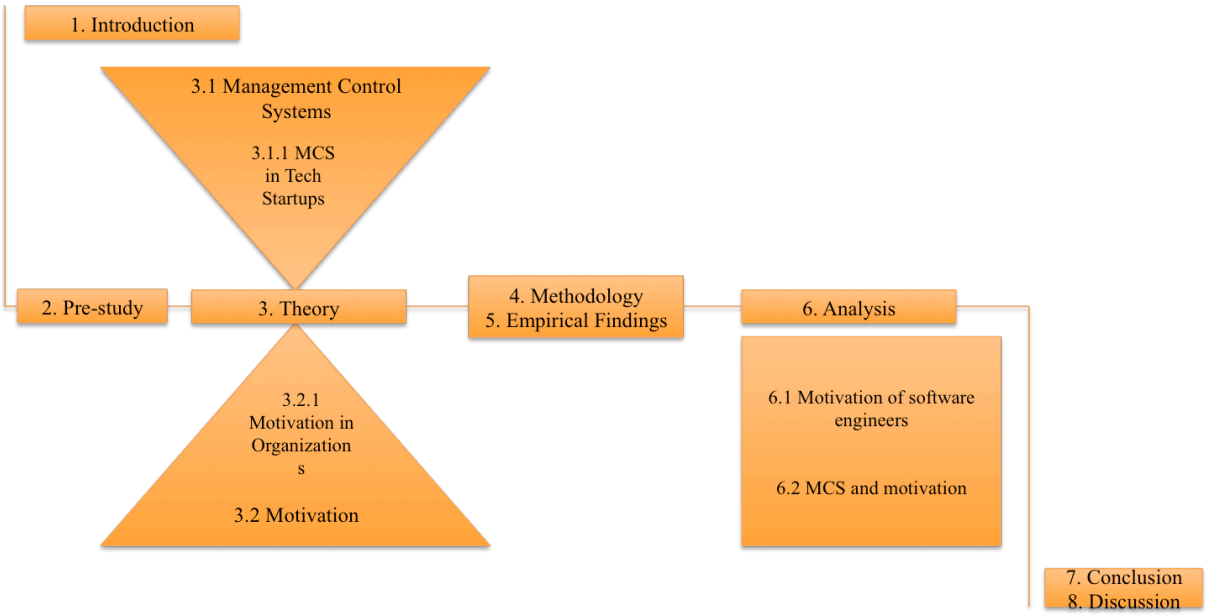

Figure 2. The thesis roadmap

2. Pre-study

This section describes the pre-study that was conducted to gain a deeper understanding of MCS and motivation in tech start-ups.

2.1. The need for a pre-study

The main reason for conducting the pre-study was to see what interesting aspects there are of how tech start-ups are using MCS. Previous studies have emphasised the importance of MCS in start-ups but little exist as to how they actually work (Davila & Foster, 2010). It is therefore of interest to understand the dynamics of MCS in practice, which is of relevance to practitioners and theorist within the management area.

The pre-study helped us distinguish our area of research and also what specific area within the organization we should focus on, namely MCS as a package and motivation of SE.

2.2. Pre-study design

The design of the pre-study took place on a firm- and individual-level to understand how tech start-ups work with MCS and then also understand how employees behave in relation to the theory of MCS. This was an iterative process and as interviews revealed the importance and challenges of aligning software engineers' goals with that of the organization, as well as keeping them motivated, the questions of the pre-study format were altered, adding questions about motivation and software engineers (See Appendix 4. Pre-study questions).

A combination of entrepreneurs, developers, consultants and acknowledged experts within the Swedish tech industry were interviewed, reason being to collect more diversified information. The method consisted of semi-structured qualitative interviews with open-ended questions, with practitioners and industry experts, which is an appropriate research method according to (Bryman and Bell, 2007). Each interview took between 30-75 minutes and was done through face-to-face or Skype, depending on availability. In total, 5 interviews were done (see Appendix 1. Pre-study interviews).

Two interviewers conducted each interview where one asked questions and the other person took extensive notes. The interviews were also recorded and each interview were discussed and summarized, first individually and then together within 24 hours after the interview, to avoid potential bias (Flick, 2009). Data was then analysed using a thematic approach to

identify patterns that would lay the foundation for the main study by limiting the scope to the most interesting parts of MCS in tech start-ups (Yin, 2003).

2.3. Findings

The first major finding is that tech departments are key for tech start-ups success. According to Cassel, General Partner of venture capital firm, Creandum, says it is software engineers who create the product that people love, and without them the company will seize to exist. Malmgren, Software Engineer Consultant at Netlight and Skatov, CTO of Starcounter goes one step further and argue for that the tech department set the pace for company growth and success, such as being able to deliver specific features required by a customer.

The second major finding, according to all the interviewed, is that it is hard to track performance and evaluate developers using traditional KPI's. Instead Skatov and Cassel mentions that managers' look more at what SE have produced (in terms of e.g. features).

Thirdly, a major finding is SE are motivated more by personal development and interesting tasks, rather than economic incentives, such as bonuses. According to Skatov, the SE is more of an "artist" and needs a "good environment" to be able to concentrate, work efficiently and grow, and if not provided this they will leave the company, no matter the pay. Motivating SE is key as they easily can switch jobs due to the high demand, according to Cassel.

2.4. Conclusions from the pre-study

Conclusively, the pre-study tells about the importance of SEs for organizations to be able to reach its goals. Additionally, individual SE are both hard to control and motivate using traditional management measures to ensure tasks are completed when due. This implies that there is a need for research to see how formal and informal MCS are used together to motivate SEs, in order for tech start-ups to reach set goals. From the findings in the pre-study we concluded that it would be interesting to study MCS as a package and motivation, in relation to SE. This led us to our research question:

How does the design of the MCS package affect the motivation (self-determination) of software engineers in the context of Swedish tech start-ups?

3. Literature Review and Theoretical Framework

The focus of this study is MCS and its relationship to motivation. This section will therefore start by examining MCS and its background (3.1) and afterwards go into MCS from a start-up perspective (3.1.1). Afterwards motivation from an individual perspective will be covered (3.2) and how individuals are motivated in organizations (3.2.1). After having identified the research gap (3.3) we will then go into the chosen theoretical framework (3.4) and develop the theoretical model (3.5) that we use to answer our research question.

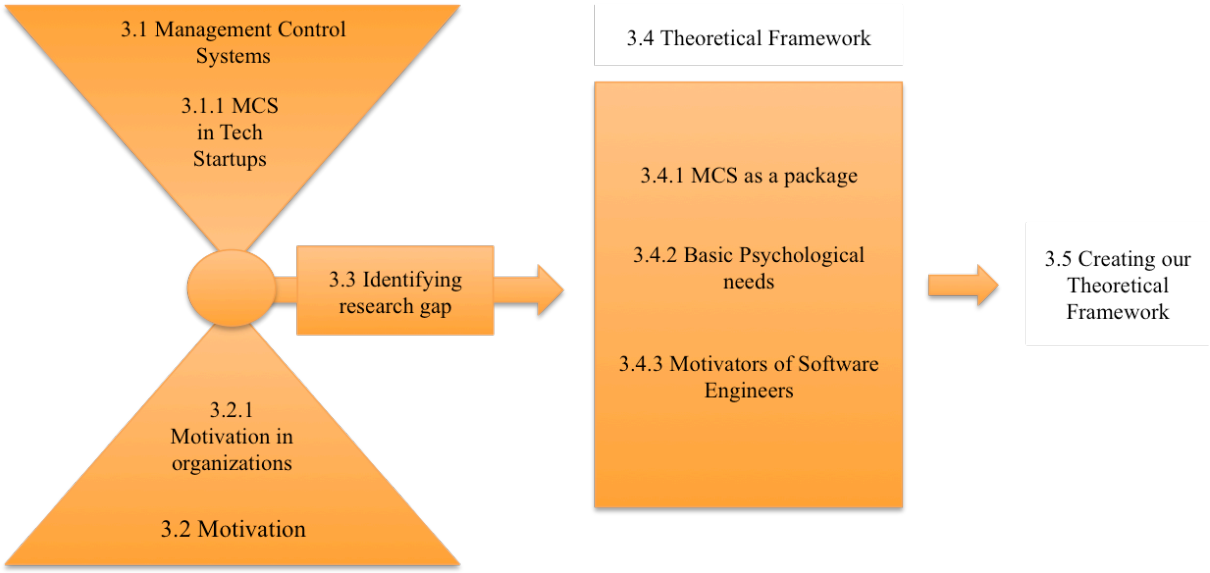

Figure 3. Theoretical roadmap

3.1. Management control systems: a review

In essence, management control starts with the managerial problem of directing employee behaviour (Malmi & Brown, 2008). In general terms, management control can be seen as the process by which management ensures that people in the organisations carry out organisational objectives and strategies and encourages, enables and sometimes forces employees to act in the organisations best interest (Carlsson-Wall, 2015). This is necessary since organisations consist of individuals with their own personal interests. Regardless of whether these persons want to help attain organisational goals, the organisation needs to integrate their efforts and direct them toward specific goals. Hence, organisations need to influence or control the behaviour of people if they are to fulfil their plans and achieve their goals (Flamholtz, 1996).

Within the area of management control, managers can use various management control systems (MCS) to direct employee behaviour. Broadly said, management control systems (MCS) support managers in their attempts to implement organizational strategy, communicate organizational goals, and align the behaviour of individuals to these organizational (strategic) goals (Otley and Berry, 1980; Simons, 1995; Langfield-Smith, 1997; Chenhall, 2003). At its inception as a separate area of study through Anthony's (1965) seminal work, management control tended to encourage a strong emphasis on financial, accounting-based controls (Otley & Merchant, 2007). But during the development of the field, definitions of what constitutes an MCS have started to vary in how wide they are, i.e if they include only accounting based controls or a wider set of both formal and informal controls. To clarify, Jaworski (1988) defines formal controls as "written-down, management initiated mechanisms" and informal controls as "unwritten, typically worker-initiated mechanisms".

So comparing the three dominant definitions of MCS, Simons (1995) defines MCS as "the formal, information-based routines and procedures managers use to maintain or alter patterns in organisational activities". This definition is thus narrowly focused on information-based routines that are formalized. Merchant and Van der Stede (2012) on the other hand defines MCS wider as "all the devices or systems managers use to ensure that the behaviours and decisions of their employees are consistent with the organization's objectives and strategies". Lastly, Malmi & Brown (2008) provides the widest definition as they say that "all those systems, rules, practices, values and other activities management put in place in order to direct employee behaviour should be called management controls. If these are complete systems, as opposed to a simple rule (for example not to travel in business class), then they should be called MCSs."

3.1.1. MCS as a package

Though MCS as a package dates back thirty years (Otley, 1980) it is not until recently research has started to provide a conceptual typology of an MCS package (Bedford et al., 2010; Malmi & Brown, 2008; Sandelin, 2008; Strauss et al., 2012). Advancement in the field have occurred since MCS as a package is a broad model that consists of several control systems, in which studies have produced unclear findings and conflicting results as too few components of MCS have been considered; instead of a more comprehensive and integrative approach (Haustein et al., 2014; Merchant & Otley, 2007; Sandelin, 2008). On the contrary, MCS as a package considers several systems simultaneously e.g. formal MCS, such as

management accounting systems, and informal systems, such as culture, and the subtle linkages that exist between them (Ahrens & Mollina, 2007).

MCS as a package have evolved into different frameworks. Simons (1995) focuses on the formal, information-based controls and emphasise the use for managers to direct strategy. Merchant & Van der Stede (2007) focus more on the design of the MCS and classify controls after their focus, either control dictate outputs (results controls), influence norms values and beliefs of employees (personal and cultural controls) or they specify behaviour (action controls). This differs from Malmi & Brown's (2008) study, which takes on a broader scope and divides controls in the categories of planning, cybernetic, administrative, rewards and cultural controls. This framework aims at creating a better understanding of how these controls are actually configured as a package across organisations, if there are particular configurations that systematically exist in specific settings and how these then support achievement of organizational goals and control activities to increase the performance of the organization.

3.1.2. Management control systems in early-stage companies

Literature is starting to surface of how start-ups develop MCS. Not only do MCS exist in start-ups, but also affect valuation, revenues and the adoption of financial statements (Armstrong et al., 2006; Sandino, 2007). Furthermore, the intensity of MCS adoption, in terms of HR-policies and stock option compensation, have shown to increase the number of employees, and MCS adoption have proved to increase as a result of venture capital funding (Davila, 2005; Davila & Foster, 2005; Hellmann & Puri, 2002).

Looking more specifically at entrepreneurial organisations, previous empirical research from for example Collier (2005) have found that informal controls such as social control, group norms, socialisation and culture were more important than formal controls. When studying entrepreneurial organisations it therefore seems crucial to take informal control into account, not only because they have not developed extensive formal MCS but also there exists a resistance from the entrepreneur to introduce formal MCS (Davila & Foster, 2010). Studies show that MCS are introduced in a sequence rather than all at the same time, and as the company grows (Moores and Yuen, 2001; Sandino, 2007). Interestingly, Sandino's (2007) study showed that early-stage companies introduce similar basic management accounting systems, while as they grow they start to adopt more complex systems.

MCS have an impact on explaining high-growth start-ups from low-growth start-ups. A study conducted on 150 small and medium Scottish firms showed that "forward planning" and "organization and systems "… have a positive impact on performance (employee growth, profitability and productivity) (Reid & Smith, 2000). Likewise Davila & Foster (2005; 2007; 2010) came to the conclusion that start-ups which introduce formal control elements earlier in a higher-intensity can have up to three times higher performance in terms of valuation, headcount and turnover. Interestingly, Davila et al. (2009) show that an introduction of MCS increases innovation in terms of product development performance. Moving on, though MCS as a package is a relatively new concept in start-ups they are both present and both consist on informal and formal control systems (Strauss et al., 2012). Studies have emphasised that the ability to balance the MCS package is crucial in uncertain environments, characterized by limited resources, dynamic market conditions, and changing stakeholders (Zimmerman and Zeitz, 2002; Zott and Huy, 2007). Strauss et al., (2012) shows that all stakeholders are central for the firms' growth and it becomes crucial to meet the demands from these stakeholders to create a MCS that can be considered balanced.

3.2. Defining motivation theories

Motivational theory is something that has been widely studied the last decades and has evolved into several different frameworks and sub-theories, depending on the context at which they are studying, e.g. looking at well-being or goal-fulfilment (Laurie M. 2005). Motivation in the context it will be used in this study is referred to as "… means to be moved to do something" (Deci & Ryan, 2000). Motivation is an *individual* phenomenon and used to predict behaviour (choices) by analysing external and internal forces, as motivation is seen to be in the control of the individual (Mitchell, 1982).

One of the theories that have gained wide acceptance in motivational theory is Self-Determination Theory (SDT) (Ryan & Deci, 2000). SDT studies the human motivation and personality by emphasising on peoples inner resources for personality development and behavioural self-regulation for predicting behaviour of individuals (Ryan & Deci, 1985; 2000). It is based on what is called an organismic dialectical approach, which means that it assumes that people (1) are seen as active organisms, which masters surrounding challenges, as well as incorporates new experiences into a coherent sense of self; and development of the individual requires (2) support from the social context, that have the power of either supporting or thwarting your psychological growth (Deci & Ryan, 2000; Deci & Ryan, 1985).

The social context together with the active organism provides the basis for SDT in making predictions about behaviour.

Table 1. The Self-Determination Continuum (from not being self-determined to completely self-determined) according to Deci & Ryan, 1985; 2000, adopted by von der Trenck et al. 2014.

Motivation Regulatory style		**Description**	**Self-regulation**
Amotivation Non-regulation		Behaviors perceived as caused by forces out of one's control	n/a
Extrinsic motivation	External regulation (ER)	Regulated by external means, no integration of external values	Very low, controlled by contingencies
	Introjected regulation (IJ)	Recognition of values by an individual, but not a complete acceptance of these values as own	Moderately low, consequences are not administered by others, but by the individuals
	Identified regulation (ID)	Recognition of the underlying value of behavior and acceptance as own, no external pressure	Moderately high, identifications still can be isolated from the self
	Integrated regulation (IR)	Inclusion of the identification, but also assimilation of the value to the remaining aspects of the self	Very high, much in common with intrinsic motivation; extrinsic motivation, as actions are executed to attain certain outcomes
Intrinsic motivation Intrinsic regulation		Activity directly refers to internal values, desires, interests	Very high, actions are executed because of enjoyment of an activity

In order to become self-determined Deci & Ryan (2000; 2008) argue that individuals need to fulfil three basic psychological needs, for effective functioning and psychological health. These are: autonomy, competence and relatedness.

Table 2. Description of the three basic psychological needs.

Basic Psychological Need	Definition	Source
Autonomy	Autonomy is the individuals innate need to feel the freedom of choice and unforced, as well as psychological freedom when doing an activity.	deCharms, 1968; Deci & Ryan, 2000
Competence	Competence is defined as a person's innate need to feel competent in its encounter with the surroundings.	Deci & Ryan, 2000; White, 1959
Relatedness	Relatedness is defined as peoples innate need to reckon themselves linked to others, to be part of a group, be feel affections and caring and caring for others.	Baumeister & Leary, 1995

According to Deci & Ryan (2008) these needs provides the means of understanding how the social forces and interpersonal environment affect motivation. As the needs are universal, SDT does not focus on the varying strength of the needs, but rather to which extent they have been satisfied through concepts of life goals and causality orientations (Ibid).

Life goals are about the interplay between different types of motivation that are affecting an individual. SDT distinguishes between amotivation and different types of motivation that

enables an outcome, namely intrinsic and extrinsic motivation (Deci & Ryan, 1985; Laurie, 2005). There is a clear distinction between intrinsic and extrinsic motivation as intrinsically motivated individuals "... engage in an activity because they are interested in and enjoy the activity [while] extrinsically motivated, individuals engage in activities for instrumental or other reasons, such as receiving a reward" (Eccles et al. 2002). People whose motivation stems from being more intrinsic have more interest, excitement and confidence, which in turn results in better performance, persistence and creativity, relative to those that are more extrinsically motivated/demotivated (Deci & Ryan, 1991; Sheldon et al., 1997). Research has shown that intrinsic motivation is facilitated by social conditions that facilitate individual's feelings of competence, autonomy and relatedness, which increases motivation, performance and well-being (Deci & Ryan, 2000 (Deci et al., 1999).

Causality orientations is concerned with *autonomous* and *controlled* motivation (Deci & Gagne, 2005; Deci & Ryan, 2008). An individual who is autonomy oriented experience social contexts as autonomy supportive and self-determined. Meanwhile, control oriented people have a tendency to experience social contexts as controlling. In effect, this is related to the perceived locus of causality (PLOC), as degrees of autonomy affect motivation on a wider spectrum: intrinsic, integrated, identified, introjected and external (See table 1), which in turn affects individuals in terms of well-being, performance and creativity (Deci & Gagne, 2005; Deci & Ryan, 2000; Ryan & Connell, 1989).

3.2.1. Individual motivation in organizations

As mentioned above, individuals are not only motivationally affected by internal factors but also external ones (Deci & Ryan, 2000). There exist several different studies on motivation in an organizational context and its effect on internal cognitive processes, such as expectancy theory (Vroom, Porter, Lawler, 1964), equity theory (Adams, 1963), goal theory (Locke, 1990), hierarchy of needs (Maslow, 1954) and two-factor theory (Herzberg, 1968). A usual approach to understand motivation is to look at *what* motivates individuals (content theories) or *how* motivation occurs (process theories) (Laurie M. 2005). SDT theory takes both of these factors into account and specifically looks at competence, relatedness and autonomy for understanding *what* (content) and *how* (process) goals are pursued (Deci & Gagne, 2005). This results in the ability to predict to which degree people are able to satisfy their basic psychological needs to pursue and attain outcomes (Deci & Ryan, 2000). This creates an important difference between SDT and content theories such as Maslow's Hierarchy of needs, and process theories such as Vrooms Expectancy based Models (Deci & Gagne, 2005).

In 2005, Gagne & Deci conducted a study that relied on data from several different laboratory experiments and field studies on motivation in organizations. Their study concluded that work climates that promote satisfaction of the three basic psychological needs increase not only workers intrinsic motivation, but also enhance full internalization of extrinsic motivation.

In effect, fulfilling the basic psychological needs have a direct positive impact on work engagement, well-being and performance evaluations (Deci et al. 2001; Bard et al, 2004). The organization influences intrinsic motivation of its employees and affect outcomes at work as persistence, performance (especially of hard, creative tasks that require a conceptual understanding), job satisfaction, and positive work-related attitude, align goals of the employee with the organization and psychological well-being (Deci & Gagne, 2005). Furthermore, Grant (2007) illustrates how work contexts motivate employees to want to create a positive impact on motivation and thus increases humans' self-determination. Likewise the fulfilment of the three psychological needs in an organizational context have an impact on performance as they facilitate self-motivation of goals valued by the organization (Bard etl al., 2004; Rich et al., 2010). Additionally, autonomy-supportive environments' (promotes perception of choice) facilitate self-determined behaviour, which lead to higher perceived competence in accomplishing desirable outcomes (Williams et al., 2000; Williams and Deci, 2001).

3.2.1.1. Motivation of software engineers

Software engineers are regarded as an important category of knowledge worker and because of the importance of these workers the understanding of their motivation is increasing (Carneiro, 2000; Witt & Burke, 2002; Beecham et al., 2008).

Frangos (1997) reflects how the work environment and management procedures can de-motivate or motivate the software engineer. For example, a lack of office space and engineer concentration, unpaid overtime, non-productive meeting cultures, performance appraisals and absence of teamwork all contribute to de-motivation. Furthermore, leader's behaviour will affect subordinates' motivation (Beecham et al., 2008; Li et al. 2006). Beecham et al. (2008) have synthesized a model of motivation in software engineering based upon reviewed literature. Within this framework motivators of individual SE, in a work context, are described and related to the SDT framework developed by Deci & Ryan (2000) by categorizing motivators depending on their extrinsic/intrinsic nature.

Moreover, fulfilling the three basic psychological needs have shown to affect individual software engineers. Koh et al. (2010) show that engineering students self-determination increases if the students perceived their basic psychological needs to be fulfilled. Von der Trenck et al. (2014) explorative case study concludes that autonomy and competence play a more important role than relatedness and speculates it can be because software engineers deep dive into a topic without external help.

3.3. Identifying research gap

While research about MCS in early-stage companies has focused on, single, formal, MCS such as accounting or HR systems (Davila & Foster, 2007; Hellman & Puri, 2002) these have however produced unclear and conflicting findings since too few components have been studied (Merchant and Otley, 2007; Ferreira and Otley, 2005; Covaleski et al., 2003; Otley, 1999). At the same time there is limited research that studies start-ups by taking an overall grip of MCS, taking both formal and informal MCS into account (Collier, 2005; Sandelin, 2008). This is important as start-ups are relying both on informal and formal MCS (Collier, 2005; Hutzchenreuter, 2009) and all parts of MCS are dependent on one another in order to work and to strive towards the same goals (Sandelin, 2008). Consequently, Malmi & Brown (2008) argues:

"Our knowledge about other forms of control (such as informal, cultural, or administrative controls) in the management control area, their interaction with formal controls, and the influence of different control packages on organizational performance is still underdeveloped".

This implies that certain frameworks that consider MCS as package are not applicable for this study. Simons (1995) framework uses only formal controls in their study, which would undermine the study in terms of not accounting for informal control. Merchant & Van der Stede's (2007) framework does provide several advantages but would also limit the research. More specifically, it neglects the organizational structure as a control measure and the framework emphasise MCS as mechanisms to realize strategy, rather than investigating the design of the MCS package and its impact on affecting strategy (Bedford et al., 2010).

Though recent studies show that MCS has an affect on motivation this research has focused on formal MCS and employees perceived PLOC (Chen et al 2014; Jorgensson & Messner, 2009), thereby neglecting informal control systems and how the basic psychological needs are facilitated. Similarly to Chen et al. (2014) motivation of SE has primarily focused on different types of motivators (intrinsic/extrinsic) and different job characteristics that exists to motivate SE (Beecham et al., 2008). It is thus under researched how both informal and formal MCS together acts an external factor which affects the self-determination of individual SE's, and thus have an impact on SE's performance.

3.4. Theoretical framework

While there is plenty of research on both MCS and motivation of software engineers, no research papers have specifically studied the intersection between the fields, in a tech start-up setting. To cover this research gap we build our analysis on three theoretical models. First we will present a model by Malmi & Brown (2008), which outlines MCS as a package (3.4.1). Secondly, we will introduce Beecham et al. (2008) model of motivators for software engineering (3.4.2). Lastly, a model by Ryan & Deci (2000) outlining the three basic psychological needs (3.4.3).

3.4.1. Management control systems as a package

Malmi & Brown's (2008) study develops a "package of control" framework. This framework starts from the idea that control is about managers ensuring that the behavior of employees is consistent with the organizations objectives and strategy. This framework is suitable because of several reasons. It excludes pure decision support systems, as these are not designed to hold organizational members accountable for their behavior. The framework also takes into account both formal and informal control systems. Lastly, it puts emphasis on the design of the MCS package and its effect on employee and organizational alignment i.e. the use of planning can help decide on future activities but it can also cause more commitment to these plans.

Five different systems are postulated by Malmi & Brown (2008), as seen in Figure 4. Cultural controls are on the top of the model as they are broad yet subtle controls. They are slow to change and therefore provide the contextual frame for other controls. In the middle of the figure are the planning, cybernetic and reward and compensation controls. These are assumed to be tightly linked in many contemporary organisations and are presented in temporal order from left to right. At the bottom are the administrative controls that create the structure in

which planning, cybernetic and rewards and compensation control are exercised. To attain a more comprehensive understanding of the different systems' and its components, see Appendix 2.

<table>
<tr><td colspan="7">Cultural Controls</td></tr>
<tr><td colspan="2">Clans</td><td colspan="3">Values</td><td colspan="2">Symbols</td></tr>
<tr><td colspan="2">Planning</td><td colspan="4">Cybernetic Controls</td><td rowspan="2">Reward and Compensation</td></tr>
<tr><td>Long Range Planning</td><td>Action Planning</td><td>Budgets</td><td>Financial Measurement Systems</td><td>Non Financial Measurement Systems</td><td>Hybrid Measurement Systems</td></tr>
<tr><td colspan="7">Administrative Controls</td></tr>
<tr><td colspan="2">Governance Structure</td><td colspan="3">Organisation Structure</td><td colspan="2">Policies and Procedures</td></tr>
</table>

Figure 4. MCS as a package according to Malmi & Brown (2008).

3.4.2. Motivators for software engineers

The literature review by Beecham et al. (2008) provides a thorough table of what motivators SEs have. The advantage of theses motivators is that they are specific towards SE in their work setting. This framework is thus suitable as guidance in relating the design of the MCS package to motivators of SE. The table provided by Beecham et al., also suits well as it will help us determine how different motivators of SE relate to the basic psychological needs. This becomes important, as it will help us see how an external factor as the MCS package affects the self-determination of SEs.

The framework of Beecham et al. (2008) concludes that there are different intrinsic/extrinsic motivators of SE. Though Beecham et al., separates motivators in intrinsic/extrinsic motivators, we will cluster them all together as they all enable ACR, though to a varying extent. In the following table the different motivators of SE can be seen, ranked according to the number of studies validating their importance.

Table 3. Motivators of software engineers according to Beecham et al. (2008).

Motivator	Description
Identify with the task	Clear goals, personal interest, know purpose of task, how it fits in with whole, job satisfaction; identifiable piece of quality work)
Employee participation	Involvement/working with others
Good management	Senior management support, team- building, good communication
Career path	Opportunity for advancement, promotion prospect, career planning
Variety of work	E.g. making good use of skills, being stretched
Sense of belonging	Supportive relationships
Rewards and incentives	E.g. scope for increased pay and benefits linked to performance
Recognition for work done	Recognition for a high quality, good job done based on objective criteria
Development needs addressed	E.g. training opportunities to widen skills; opportunity to specialise
Technically challenging work	Solving complex problems
Job security	Stable environment
Feedback	Recognition (for a high quality, good job done. This is different to 'rewards and incentives' which is about making sure that there are rewards available).
Autonomy	Being able to take own decisions
Good work life balance	Flexibility in work times, caring manager/employer, work location
Making a contribution	Degree to which the job has a substantial impact on the lives or work of other people
Empowerment	Personal responsibility
Appropriate working conditions	Good equipment/tools/physical space/quiet
Trust/respect	Being treated fairly
Equity	Ownership of the company
Successful company	Financially stable company
Sufficient resources	E.g. able to pay salaries

Furthermore, Beecham et al (2008) shows that the most widely reported outcomes of motivated SEs' are improvements in productivity, project delivery time and improved project success. As such, the outcome of increased motivation i.e. performance of SE will be referred to in terms of these three factors.

3.4.3. Fulfilling needs to fulfil goals

The basis for choosing the SDT framework proposed by Deci & Ryan (2000) amongst all different types of motivational theories is that it primarily takes basic psychological needs into account, as well as the emphasis of external influences on individual motivation.

Furthermore, these needs also directly related to the intrinsic/extrinsic motivators identified in Beecham's framework.

The basic needs that enable motivation are: *autonomy*, *competence* and *relatedness*. The degree to which these needs have been fulfilled (based on the motivators acting as guidance by Beecham et al. 2008) will determine the degree of self-determination of individual SE's. In effect it will strengthen the analysis and increase the understanding of how the design of the MCS package is related to the self-determination of the individual SE's.

3.5. Forming our theoretical framework

To be able to address the research question (Figure 1), we will use a theoretical framework where we will be able to combine models on Management Control Systems as a package and motivation. The framework presented in Figure 5 illustrates how MCS affect motivation of individual SEs and thus performance, using the frameworks of Malmi & Brown (2008), Deci & Ryan (2000) and Beecham et al. (2008).

Malmi & Brown's (2008) extensive framework will be used to explain how the design of the MCS package acts as an external factor to influence the self-determination of an individual SE in a work context and thus his/hers performance (as identified by Beecham et al. 2008). The basic psychological needs (ACR) take the form of different intrinsic/extrinsic motivators, as identified by Deci & Ryan (2000) and Beecham et al. (2008). We will thus use these motivators as guidance to categorize motivators under the basic psychological needs in order to see how the MCS package affects the self-determination of individual SEs.

As self-determination is dependent on external factors, it is proposed that MCS as a package affects the self-determination of individual SE, rather than vice versa. The same relationship is proposed between self-determination and performance as an increase in self-determination increases performance (Bard et al. 2004). Together, these frameworks are seen as the most suitable to address our research gap.

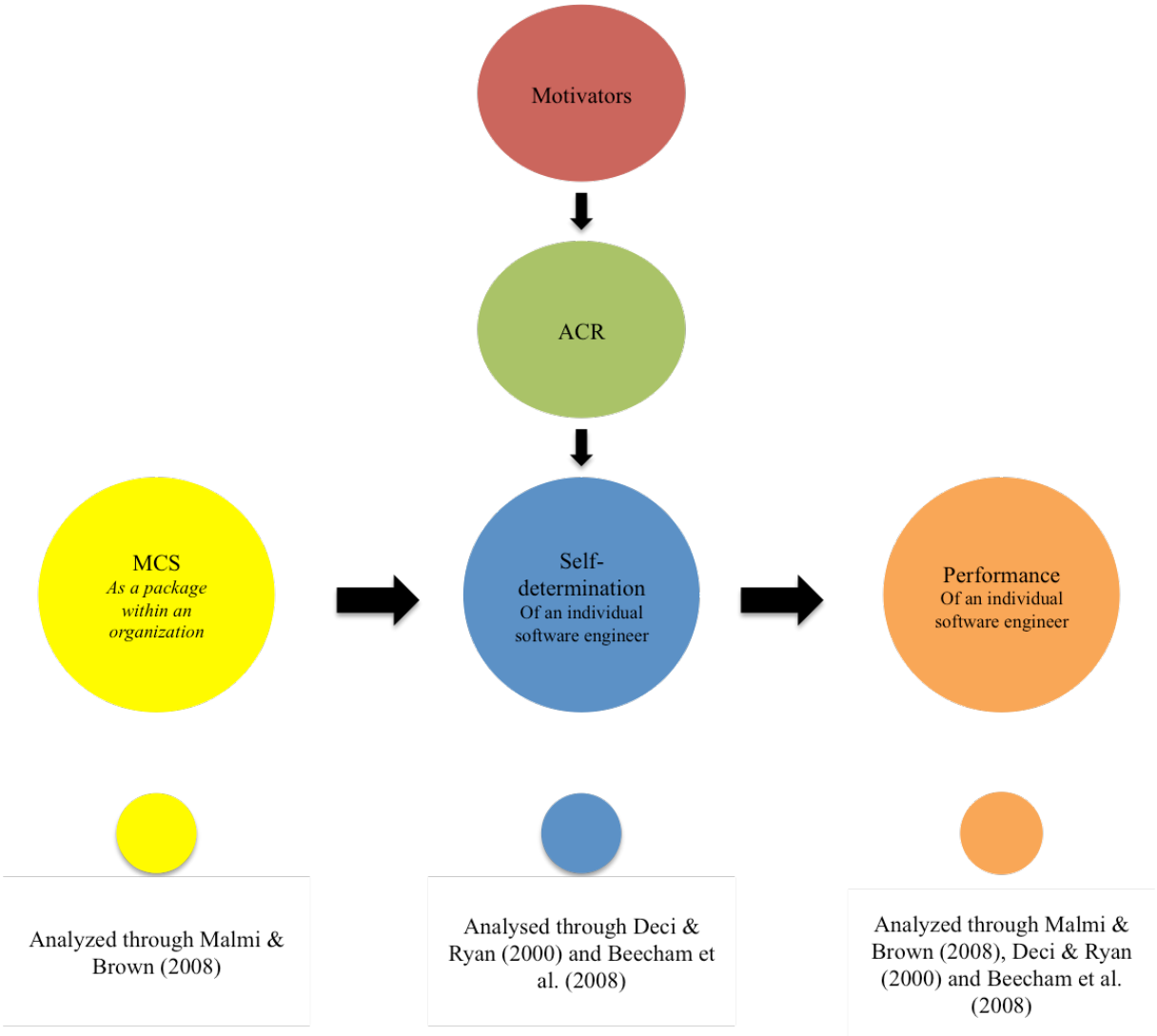

Figure 5. Theoretical framework of this study.

4. Methodology of Main Study

This section will present the methodological choices that were made in designing the study and analysing the data. The section consists of six chapters that describe the research approach and its reliability, validity and transferability.

4.1. Methodological fit

Prior research has suggested that there is currently a lack of understanding of the dynamics of MCS as a package (Strauss et al., 2012). Specifically looking at MCS as a package within a start-up and how its affects the motivation of an individual (Chen et al. 2014). Therefore an explorative study is of value as it allows the study to map out were it would be of value for researchers to conduct further research (Bell, 2006). Due to the extensive and complex nature of MCS as a package the study will aim to identify patterns using a thematic analysis (Edmondson & McManus 2007).

A qualitative approach was chosen as it is the most appropriate when it comes to engaging in an explorative study and the literature of a topic that is in a relatively nascent state (Edmondson & McManus, 2007; Yin, 2003). Though a quantitative study would make the results more transferable a qualitative one creates an understanding for individual perspectives and patters of behaviour, and evaluating the applicability of theoretical models (Flick, 2009; Trost, 2010). It is also more commonly used when going in-depth into a topic, were the ambition is to identify concepts and ideas (Ghauri & Gronhaug, 2005).

4.2. Research approach

The study was guided by existing theory in MCS and motivation. However, unlike deduction that starts from theory and induction that starts from facts (Bell, 2006), this study will use an abductive method (Patel & Davidson, 2011). Reason being the nascent state of the research area and using an abductive approach combines theory with practice, which makes it possible to identify patterns for further explanations (Ibid)

The study was conducted by first using mapping out MCS theoretical frameworks. To ensure that the approach is relevant and the most interesting aspects is studied a pre-study was conducted. Afterwards we chose the scope and collected empirical data from carefully selected case companies using in-depth semi-formal interviews. Based on a thematic analysis we evaluated the existing theoretical frameworks applicability and developed them (Braun & Clarke, 2006).

Figure 6. The process of the study.

4.3. Case study

Case studies are an appropriate research method within a real-life context where no objective metrics exists as it allows one to understand "how" and "why" questions (Yin, 2003; Van Lehn, 1989). Additionally, Yin (2003) argues that a case methodology is appropriate if the researchers have limited control of the events that take place. By using one or more cases it allows researches to "create theoretical constructs, propositions and/or midrange theory from case-based, empirical evidence" (Eisenhardt & Graebner, 2007, p.25). However, some scholars argue that case studies only contribute to creating hypothesis and that they are too specific (Abercrombie et al., 1984). This have however been argued against by several authors that argue for the use of case studies as they provide a rich understanding and is a valid research method (Flyvbjerg, 2006).

4.3.1. A multi-case study

The reason for conducting a multi-case study instead of a single case study is to provide a broader collection of data and thus increase the validity of the findings (Yin, 2003). By using multiple sources it increases the likelihood and possibility to identify general patterns of the research area and scope; single case studies are harder to generalize (Eisenhardt, 1989; Yin, 2003).

It could have been beneficial to do a single case study, as it would be possible to find more underlying relationships between MCS and motivation (Yin, 2003). However, as the research scope revolves around individuals in one department of a start-up, it was hard to reach sufficient number of people to take part of the study to see any general patterns. It was thus seen as better with a multi-case study to reach data saturation but also to find more generalized patterns. Of course, the data collected could have been complemented with surveys to strengthen the transferability of our results. This was however not deemed as possible due to the time frame that existed.

4.3.2. Selection of companies

Tech start-ups have a quite loose definition and it is therefore important to use companies that are as similar as possible to be able to identify clear patterns.

All chosen case companies were active within the IT industry. The reason for this is because of its increasing relevance in the economy and their dynamic nature (Paternoster et al. 2014). Moving on, all start-ups have received venture capital funding as it has shown to increase the adoption of formal MCS (Davila & Foster 2005). To limit differences that can exist due to culture (Malmi, 2013) and simplify data collection, all selected companies have their headquarters in Stockholm or close proximity to it. The same goes for the different VC companies that have invested in the tech start-ups. Most studies on MCS in start-ups have used samples ranging from 50-150 employees, as it is at this stage that start-ups attain formal MCS. However, both formal and informal MCS can be present in smaller start-ups (Sandino, 2008; Strauss et al., 2012). Additionally, Davila & Foster's (2007) study show in their graph in Appendix 2 the correlation between VC funded start-ups and the intensity of MCS adoption. Within this graph the curve is at its steepest around 10-30 employees. Due to this the companies that have been chosen for our sample has a range between 10-50 employees, in order to further add to the MCS research that Davila & Foster have built on.

We started to e-mail potential companies that were within the Swedish IT-sector and having received VC-funding. We used The Swedish Private Equity & Venture Capital Association and VC firms that were located in Stockholm as our sample. We contacted companies that existed in their current portfolio and used contacts that were provided from our pre-study.

Table 4. Companies that were contacted

	Number of companies
Companies that were e-mailed	137
Companies that did not reply	108
Companies who did not want to participate	13
Companies who wanted to participate in the study	16

Based on the companies that wanted to participate in the study, we wanted to ensure that the companies were as alike as possible in terms of technical platform, size and age. All companies chosen are therefore ten years or less, which is similar to the sample Davila & Foster (2007) used when analysing early-stage start-ups. There is one company in the sample that is older. However, in this case the software engineering team has been split up in two parts due to launch of a new product that was launched in the beginning of 2016. As we are

looking at SE specifically it is therefore seen as viable candidates in the study. In total, four companies were selected as case companies, were three of the companies sold their services in the form of software and one in the form of a combination of software/hardware. In the best of worlds it would be beneficial to have two of both but we were unable to achieve this in the study without compromising on the other factors we have chosen from. To add to the confidentiality of the study and create more open interviews, all companies and employees were made anonymous.

Table 5. Selected case companies.

Company	Line of Business	Founded	# of Employees	Type of Platform	Revenues 2014/2015 (TSEK)
A	Online Marketplace	2011	23	Software	16 268
B	Recruitment software	2008	45	Software	17 500
C	Indoor positioning system	2010	16	Hardware	4 768
D	Online collaboration tool	2002	35	Software	37 212

4.4. Data collection and documentation

Data was mainly collected using in-depth semi-structured interviews. During the interviews we were also provided internal documents such as organizational structure that were compared to the collected data.

4.4.1. Choice of people

The interviewee sample was established to provide a rich understanding of the research gap in Figure 1. To enable this we talked to the companies we established contact with and discussed the study and which people who were involved in managing software engineers. This could vary between companies but based in the information given we identified the following organizational members most worthy of interviewing:

1. Chief Executive Officers (CEOs)
2. Chief Technological Officers (CTOs)
3. Chief Product Officers (CPOs)
4. Technical Leads (TLs)
5. Software Engineers (SEs)

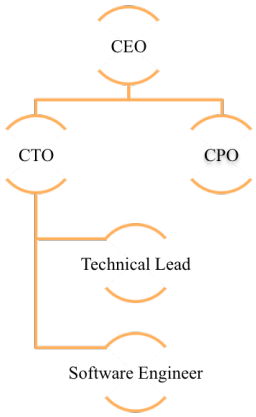

Figure 7. Interviewed organizational members within a hierarchy tree

The data sample consisted of 18 interviews, with 4-5 interviews per company. The interviews took between 45-75 minutes (Appendix 3) and took place on the location preferred by the interviewee. Seven of the interviews were conducted through Skype and the rest took place face-to-face. Looking at each specific group, four CEOs were interviewed, three CTOs, three CPOs, two technical leads and six software engineers.

4.4.2. Design and documentation of interviews

As we wanted to gain as rich understanding as possible about the dynamics of MCS and its relationship to motivation, we interviewed people from different functions and organizational levels, as well as used provided documents from the interviews (these documents were however secretive which means we could not include them in the appendix).

The questionnaire was a result from the scoping in pre-study and based on existing theory about MCS and motivation. Because of the relatively unknown phenomena, only the general aspects of the study's design were decided upon in advance (Yin, 1993). The questionnaire was divided in three parts: (1) the interviewee's background & Company goals, (2) questions about MCS and (3) Motivation.

Semi-structured interviews is recommended by Bryman and Bell (2007) as it enables the interviewee to explain his/her account without being primed by the question bias, which is why the questions also was open ended; this enabled the interviewee to expand on topics that were discussed during the interview (Gillham, 2005). Due to the abductive nature of the paper, iterative findings that were made during the data collection led to some questions being emphasised more to certain groups than others. Moving on, to avoid misinterpretation and ensure that the interview was interpreted correctly, we summarized and clarified what the interviewee said during the interviews (Flick 2009). Moving on, two interviews were

conducted in English and the rest in Swedish, which entails translation and a risk of losing underlying meaning. To avoid any misinterpretation the translations were therefore rigorously checked (Flick, 2009). See appendix 5 for the interview questions that were asked in the main-study.

As in the pre-study, each interview took place by two interviewers, were one took extensive notes while the other conducted the interview. Within 24 hours of the interview it was discussed and summarized, together and individually, to reduce any potential bias (Flick, 2009). Additionally, each interview were recorded and crosschecked with the tapings to add missing quotes or themes in an interview.

4.5. Analysis and interpretation of data

To analyse the data we used a pattern matching method suggested by Yin (2003) to enhance its quality. This implies comparing theoretical and empirical patterns in interactive matter to draw conclusions (ibid). The analyses were thus made through colour coding notes by focusing on phrases, key words and themes. A four-step analysis was applied: first, each interview with SEs and their motivators was analysed separately and then analysed together by looking at common patterns and themes. Secondly, each company's individual MCS package were analysed by looking at relationship between the controls of the MCS package. Thirdly, each companies MCS package were then analysed in relation to the motivators of individual SEs, found in step 1. Forth, each company's MCS package, the relation between its controls and their mapped effect on the motivational factors was compared in between companies to analyse possible common patterns that would strengthen the generalizability of the findings.

To further enhance the study we did the four-step analysis on an individual basis first and by doing so were able to compare our conclusions and make sure we were not affected by each other's initial interpretation (Flick, 2009).

4.6 Data quality

While there are number of ways and methods to make sure that the quality of a quantitative study holds specific standards, it is not as developed when it comes to a qualitative one. In this study we have choose to build on the traditional criteria of reliability and validity. But as the two factors alone do not provide an adequate quality assessment in a qualitative study, we have decided to add transferability to increase the quality of the data (Flick, 2009).

4.6.1 Reliability

Reliability depicts the quality of the measurements used in the study (Trochim & Donnelly, 2008). In order to increase reliability a pre-study was conducted (Flick, 2009). Furthermore, Flick (2009) emphasise the importance checking the dependability of procedures and data of documentation of the conducted interviews, by clearly distinguishing between the difference of interpretation of the researcher and what was said by the interviewee, to make sure comparability between interviews are achieved. Recording each interview ensured dependability, but also taking independent notes that was later compared to each other. During the research we tried to be as structured as possible with our data, and storing all notes, transcripts, recordings in order to verify and confirm what conclusions that was made (Bryman & Bell, 2007).

4.6.2 Validity

Validity looks at weather investigators actually observe what they perceive they do (Kirk & Miller, 1986). It therefore is of importance to view both production and presentation of data to ensure they are not biased. As according to Flick (2009) this was done by using open-ended question, trying to avoid sharing information from other interviews and giving interviewee time to respond without interfering to as large extent as possible. A limitation with using retrospective verbal reports is that there is a risk that the interviewee will reconstruct what has already taken place; a way tackling this was to pose similar questions to multiple interviews (Flick, 2009).

Moving on, data was first analysed independently in order to avoid making the interviewers influence one another (Silverman, 2013). Also, during the analysis we avoided in taking only the viewpoint of what one person said, if not validated by any other interviewee from that company and to ensure that our explanation of the findings is a valid reason we maintained contact with experts from the pre-study in order to ask for validation and potential input (Huitt, 1998)

4.6.3 Transferability

Finally, transferability concerns with if the result can be used in contexts other than that of the specific study (Bryman & Bell, 2007). In this study we tried to create transferability trough two primary ways: First, we have tried to give as thorough description as possible in the empirics of each case, as suggested by Lincoln & Guba (1985). The second way is the use of a multi-case study. By using replication logic we hope to increase the transferability of the results (Yin, 2010).

5. Empirics

This section depicts the raw data that was collected from the four case companies. These empirical findings are structured around the MCS that exist in each company and how software engineers are motivated. For a summary of the findings please see Appendix 6.

5.1. Company A

”Motivation and engagement are the most important things I have to work with” - CTO

5.1.1. Administrative controls

Looking at the IT division in company A, it consists of a small development team that are managed by the CTO who is responsible for their performance. The CTO works together with the CPO who then is responsible for how the product should be developed. These in turn report their progress to the rest of management every second Monday.

The CTO sees his role as a gatekeeper towards other departments so engineers are protected from disruptions. An engineer states “the CTO explains why things take time to other stakeholders so the rest of us can focus on our own”. There are formalized job descriptions for engineers, but since the roles are very broad they are rather determined by what tasks you are currently responsible for, which in turn are based upon their interest and qualifications.

Engineers utilize two formalized processes to guide their work. Firstly, they utilize a project management tool where tasks are sorted into the categories to do, in progress and done and tagged with who is working on them. Additionally, the team together continually update a ”DevBook” consisting of guidelines how to write code in their desired way and what is important to think about in the team.

”We try to be flexible in our way of working based upon what the team feel that it needs” – CTO

Additionally, quick meetings are held every morning with engineers and CTO to let the team know of each others progress and problems. Moreover, a retrospective meeting is held every second week to evaluate their work process and a planning meeting is held every Monday.

5.1.2. Planning controls

Planning in the organisation takes place on different levels. According to the CEO, the current long-term goals are to "conquering a new market and becoming the largest player in the existing one [by number of transaction or number of customers]". Plans, for each department, of how to achieve these targets are set in regular meetings with top management.

But according to the CTO, the development team does not work much with long-term planning: "We rarely have deadlines and long-term plans. We have a vision, but often things come in from left and right all the time [...] that could not have been predicted in advance". According to one of the engineers, the vision is to "give auction houses the possibility of accessing the Internet and simplify the buying process for end consumers".

"Before we had more lucid planning but in recent times we have stepped away from this and I do not know why [...] it has affected us in that we are not strictly following the roadmap and I am not as sure as to were we are heading. However, it does not affect me as an individual" - SE

However, engineers' plans short term in two weeks sprints – a concept from agile development – where the CTO and CPO create a prioritization list of tasks in the project management tool by taking input from customers and other departments. Engineers then choose tasks, decide how to solve them and can voice opinions on the prioritizations or the tasks. After the two weeks, retrospectives are held were the development team discusses how far they have come and what problems they face to see if everybody is on track. If an acute problem arises with the platform, these tasks are put on hold for the more stressing matters.

Deadlines for these tasks are set informally as the engineer usually talk to the CTO to managing expectations of how much time that is needed on a specific project versus how much time that can be spent. There is also a stand-up each morning were everybody in the development team updates one another of their current status, what they have done and if anything acts as a barrier.

5.1.3. Cybernetic controls

Within the development team, it is very hard to evaluate the performance of each specific engineer, as there many different values determining what is "good output" according to the CTO. An important part of measuring engineer performance takes place by the bi-weekly

retrospective were everybody gets to tell what they have done in the sprints and what they are going to focus on further. The performance of the overall product in tracked in real-time on a dashboard in the development room, looking at measures such as bugs, transactions and number of customers over various time periods.

"A lot of the performance evaluation takes place through dialogs" - SE

According to the CTO, he has distributed his time both on an operational level and an organizational level to better understand the engineer's performance. As said by the CTO: "There many different numbers about what is good output [...] and that is why I spend 25% of my time in the engineer role and the rest within management and administration". By doing so the CTO is able to get a feeling of peoples performance. There is also is a common understanding within the entire team that if someone is productive or not, which is shared by both the CTO and one of the engineers.

"Everybody has a feeling if someone feels productive or not" - CTO

One of the engineers also suggest that there is a form of automatic peer review that takes place were people within the development team are looking over each others code before it is launched to ensure quality, which in turn makes the code quality self regulated. Another form of peer review is that there are often two engineers that work on the same task. The CTO also says that there are talks each month between the CTO and each engineer to see how each person is developing, setting up personal goals and to see what can be improved. According to one of the engineers, however, these evaluations could be clearer in what is being evaluated, rather than that of a "co-worker talk".

5.1.4. Reward and compensation controls

According to the CEO no one have any stock options except for the founders and engineers state that they receive their base salary, standard vacation days as well a phone and gym card.

As the job market for engineers is highly competitive and start-ups low on capital the CTO needs to satisfy engineers by additional means. This have resulted in a budget for each engineer of 10 000 SEK to develop their skills further and 20 % laboratory time, in which engineers can do what they please as long as it has something to do related to the company.

"Its hard to connect rewards to performance and we do not have options as everybody gets their base salary and did not go into a riskful project from the start" - CTO

5.1.5. Cultural controls

"The culture in the development team is characterized by quality, high ceilings, understanding, not working in silos and community" – SE

The values of the company have just been formalized two weeks ago and the CTO reveals that decisions are not really based on them yet. An engineer reports that there is a frustration that the rest of the company does not have as clear culture as the development team. Both engineers states that the team have an ambition to constantly improve how they work and work more with soft values whereas other departments does not think of improvement in a systematic way.

"We put 80% of the time on work and 20% on reflection about how to develop" – SE

5.1.6. Motivation of software engineers

Both engineers point out that they are motivated by the great freedom that exists in their roles as they are allowed to decide themselves how they want to work, both in terms of processes and how to solve problems. Additionally, the freedom also relates to how they are controlled by management and one engineer says, "we get done when we get done and management mostly understands that".

"Software development is a very creative task. It's a form of craftsmanship. You have to have some type of artist in you. If you were to limit people in how they work it would get boring. You won't grow without freedom, but at the same time you need some frames to guide you" - SE

One engineer emphasize that team in itself is highly motivating as it "creates an atmosphere were everybody believes in the product and works very professionally to make it better". The other engineer partly agrees by stating that solving challenges and delivering a solution to customers is motivating as he feels like he is making an impact.

Both engineers mention that motivation is affected negatively if work becomes too ad hoc without clear directions, structure, frames and if other departments disturb their work by asking for favours.

5.2. Company B

"Everything in IT is about priorities, it's about knowing what's most important right now to get to the end goal" – CEO

5.2.1. Administrative controls

Engineers are split into three teams with three to four in each. Each of these has a technical lead that is responsible for their performance. The lead then report to the CTO who currently also holds the role of the CPO and is part of top management. The reason for this is that since the product is still under development they need to have the roles together to avoid communications issues.

"I work as a hand-break in both ways in the organisation. Management wants as many features as possible. Engineers want as good code as possible. My role is to balance these two demands" – Technical lead

In terms of pre-specified procedures, engineers work according to an agile methodology in their sprints, which represents itself as a dashboard in a project management software that everyone has insight into. This contains all different tasks grouped into the states not started, in development, ready for testing, and shipped. The code reviews by peers to test features ensures that engineers work according to the informal standards of the team. Their daily collaboration also controls that engineer's work according to group standards. What further limits behaviour is how the feature they are building is specified.

All engineers are "full stack", meaning that they are able to work with almost anything. Therefore, work descriptions are not formalized so one engineer is working only with one specific area. Rather, engineers focus on one or maximum two areas depending on the current needs of the company and their personal interests.

We don't check how the product's performing. We check how we can make new features faster" – Technical lead

Additionally, all engineers take part in a retrospective meeting every second week where their work processes are evaluated and possible issues within the team are discussed in order to improve their procedures. Engineers are expected to raise questions or issues of their own.

5.2.2. Planning controls

According to the CEO, the company have two fixed three-year goals in terms of users and revenue. To achieve these goals, management develops a business plan that is iterated each year. This is reliant on the product roadmap where management decides how the product should be developed and consists of three large milestones/features with 3-4 months in between.

”Before it was too task oriented. Five features were broken down into 200 to-dos and assigned to engineers. Sometimes they got drowned in stuff. Now I just tell them ’this is what we need to do to achieve the goal’ and the teams breaks it down. This way I leave the lower level higher autonomy” - CTO

The short-term planning for engineers occurs each month in a meeting led by the CTO/CPO where features that should be built in order to complete the larger milestones are prioritized. Sales, marketing and operations participate in this meeting to facilitate prioritization as they have closest contact with customers. Sometimes engineers are invited in order to set more reliable time estimates for the features.

”I talk a lot with our users. Based on that I weight what gives most value compared to how long it takes to build, then I decide what to prioritize” – Technical lead

The technical lead is then responsible for setting up two week sprints for his team where the engineers work with the prioritized features. Engineers get to choose what task to work with from this prioritized list and how to solve them as they are very unspecified – the ones that are left over are delegated. After two weeks a retrospective is held where the team evaluates what could be done better, if anyone has any problems and how to continue.

”It’s easy to get interrupted by some urgent task. All of a sudden 3 weeks have gone, you work on five things and nothing is done. This often happens when engineers are multi-tasking. Our job is to give them less things to choose from” - CTO

5.2.3. Cybernetic controls

"We evaluate quality on what you delivered in relation to time. It's a trade-off. It's easy to deliver on time if it's bad. We try to have a balance there" - CEO

Performance within the software development teams are largely measured based upon if they are done with their respective tasks when the sprints are done. But if an engineer is not able to finish it is important that he is transparent as soon as possible since delays can depend on a variety of different factors. Engineers is thus evaluated in daily stand-ups where the development teams meet for 10 minutes every morning to brief each other on their respective progress. Here the tech lead gets an overview of how progress is in relation to the schedule for the sprint.

"Because of our work structure you see really well if one works or not" - CTO

The engineers also engage in co-reviews of their peers' work where they scrutinize the features in order to quality check for potential bugs before integration. The CTO also conduct informal annual reviews where an engineer provides feedback on one or two colleagues. Engineers are also evaluated continually by looking at how they function within the team and if they help others. As the team lead says, "it's more important that you help your team than to focus on yourself". A retrospective meeting is then held every second week where the development team and the CTO evaluate the overall performance of the sprints, concrete things that has gone wrong and if something needs to be changed.

The CTO further states that he evaluates performance based upon his perception of an individual engineer's productivity. This is largely because as engineers work in teams you can tell quite fast how they perform based upon talking to other members in the team.

5.2.4. Reward and compensation controls

The CEO shares that "engineers are given a fixed salary, and that's it", but they are also informally rewarded by giving them recognition and room for personal development. Celebrations are held for all departments when a feature is launched to show the engineers behind the effort. There is also "show and share sessions" where engineers get to demo their creations for other departments, which also provides them with insights into engineers work.

Additionally, engineers are also encouraged to spend the Friday afternoon on something of their own as long as it is related to the company in some way.

"A good salary is important, but since you spend so much time at work the important thing is that it's challenging and fun" - SE

5.2.5. Cultural controls

According to the CEO, the formalized values in the company successful, quality oriented, open-minded, revolutionary and exciting/entrepreneurial and are mainly used in recruitment purposes to ensure a cultural fit.

"When you grow management looses the ability to control what happens in the teams, therefore you need other processes such as a culture that helps control employees indirectly" - CEO

The technical lead explains that he has forgotten most of the formalized values. He rather believes that what characterizes the culture in the IT department is the homogeneous team composition consisting only of white males from Sweden. As IT is very isolated and not so dependent on other departments they have their own culture that is further characterized by being very nerdy and having a deep interest in coding.

5.2.6. Motivation of software engineers

"It's motivating for engineers to show what they have done and get some applause and recognition" – Technical lead

The technical lead states that he becomes motivated by getting good feedback on a feature that he or his team has completed and being challenged in his work. It is further described how he previously changed job because he felt that he was not developing himself. Both engineers additionally emphasizes the importance of having a lot of autonomy in their work in order to feel motivated and states that it would be very frustrating if someone tried to control them too much. Conversely, freedom in how they writes create features is very important and their ability to take own initiatives for new types of solutions are thus important.

"I want interesting tasks and ability to make own choices. Tell me what you want to achieve and the goal, but don't tell me how to solve it. Then they're out on thin ice and might just as well outsource me. I am a problem solver, and if I can't solve problems I get nothing from it".
– SE

The same engineer refers to how the team acts motivating as they often come with interesting suggestions for problems and enables him to learn from them as they have knowledge that he does not have. The higher purpose within his work is also important and he describes that it feels like there's a "sensible point to what I'm doing".

5.3. Company C

"Motivation of engineers. It's a big problem. It's easy that they end up in a closed room where information neither gets in our out" - CEO

5.3.1. Administrative controls

The development organisation consists of one CTO, one CPO and five engineers. According to engineers roles are not formalized but rather defined based their competences and own interests. Subsequently, they are allowed to choose tasks themselves based on these. The CPO acts as a filter between the engineers and the sales people to allocate resources since sales have the input from clients on what features are demanded. Thereby the CPO makes sure that engineers focus on the right features whereas the CTO are responsible for the day-to-day performance of the development team.

"More customer contact would be nice. It's fun to see more of the reality than just sitting at your desk and finishing notes and tasks" - SE

Engineers work according to a process called Kanban where they have a whiteboard on the wall with all tasks are grouped into the categories back-log, selected, in progress, testing and shipped and assigned to each individual. Each person is only allowed to work on a certain number of tasks at the same time. The team has daily meetings where they updating each other on progress of these and bi-weekly meetings updating the rest of the company. Retrospective meetings to evaluate work processes in the team are held rather ad-hoc and one engineer would like to have more of this. When a feature is done, they test it internally, then

they demo it for each other and then it is demoed to the CTO who determined if it is approved.

Coding guidelines are also developed in order to ensure that engineers write their code in a desired way. Furthermore, certain processes, e.g. for launching a specific feature, are continually formalized in order to codify learning and simplify for other engineers in the future.

"The right stuff should be formalized. It is good to have some control based on informal communication and norms since if it's too formal it feels as if you're just a tiny part of a big machine" - SE

5.3.2. Planning controls

Top management has set a three-year goal in terms of revenue and has broken it down into 20-25 milestones consisting of smaller activities they need to complete to achieve it. Some of these milestones fall within IT that, as the CEO states, "is always one step behind as their demands come from clients, other departments or owners".

Based on these demands and milestones the product owner (CPO) develops a product roadmap for the coming three quarters consisting of larger features with start and completion dates. This is updated each quarter. The CTO and engineers are involved in this process in order to get better understanding of potential problems and time estimates. Each feature is then broken down into several user stories by the CPO and is inserted into their project management tool. These stories are then prioritized and time estimated in a meeting every other week with the CTO, who has more in depth view of the workload and capacity of the development team, and engineers. Here engineers get to provide input on the stories if they believe they should be done in other ways. When this is done the tasks enter their Kanban board where the day-to-day work of engineers is structured.

"In this planning phase, the CPO decides what should be done while the engineers themselves get to decide how it should be done" - CPO

What is important to take into account is that you have a fixed amount of resources and this needs to be allocated to the features that can provide the most "bang for the buck". When this

time has been allocated for each engineer, one also need to take into account that his time will be needed to split in between product development and technical bug fixes.

5.3.3. Cybernetic controls

"We don't set real goals. Engineers do the best they can and set an estimate on how fast they can work. I trust that they do their best" - CTO

Engineers are evaluated based on their adherence to the deadlines for their respective user stories and the quality of their work. But a engineer explains that if you don't keep a deadline nothing really happens since you have a continuous dialogue at the morning meetings where the progress of each engineer is evaluated through the Kanban board and each engineer can voice problems or such that can cause delays. As the CEO describes, "when you are developing a new product it is hard to exactly predict how long time everything will take since you encounter new problems along the way". The CPO and engineers describes that it is easy to evaluate performance in the development team as one can look at the Kanban board and their project management tool to see progress for each feature and person.

When a feature is done its quality is tested through integration in the customers system and feedback is provided to the engineer. If it works without triggering any bugs and the customer is satisfied one is to consider it a job well done. Additionally, before the launch of a feature code reviews are done by peers to evaluate its quality.

"Everything you measure can be manipulated. If you start measuring code, engineers will stop helping each other and it destroys the team spirit. I rather evaluate their work informally" - CTO

The CTO describes that he always has a feeling for how engineers perform - how much bugs their code produces, how much they help each other and how fast they work. He gets this by having an in depth knowledge of coding, sitting in the same room as engineers so he can see how they are doing and "hanging out" with them on a daily basis.

5.3.4. Reward and compensation controls

The company provides a base salary that's adjusted slightly to experience and a stock option program which one engineer believes improves incentives since "if you work hard you can cash out later". Besides that he believes that there are not so much structured rewards and one

rather gets positive feedback when one have completed an important task. The other engineer describes that the recognition they get at the meetings every second week where they get to show others in the company what they have built is a form of reward.

5.3.5. Cultural controls

According to the CEO and CPO, the company's formalized values are challenging, insightful and humble. The CTO does not remember these but instead refer to competence, transparency, efficiency and open-mindedness as the values that characterize the development team. The reason for this, he explains, is that the nature of their work requires it, especially as they are a very small team where people should be encouraged to test new things and make mistakes as long as they always are transparent.

One engineer thinks that his ability to voice his opinions with the CTO and CPO signifies the culture. Another identifies the culture with their product features, being premium, market leading and customer-oriented. Instead of being taught some values formally, he believes that socializing with other employees spreads the culture continually.

5.3.6. Motivation of software engineers

Both engineers emphasizes that their ability to have autonomy in how to solve problems and create features is important for their motivation. Additionally, they believe that personal development is key for their motivation, which comes from both building the company and solving the more complex problems that come along with it as well as being given time to learn new techniques that are crucial to solve them.

"We have great freedom in how to solve tasks. We get full freedom in how to implement stories as long as we follow various standards" - SE

One of the engineers further explain that what motivates him is his ability to affect the direction of where the product is going. What are important for this are the ability voice own opinions for how the planning and tasks looks like to the CPO. He continues by saying that the weekly demo sessions also are motivating as co-workers gives him recognition for his work. It is further pointed out that positive feedback from customers handed to him by the sales people has a motivating effect.

"During the spring there has been a lot of bug fixes, which meant that the focus has been quite bad for most of the engineers" - SE

It is further emphasized by both engineers that what mainly causes demotivation are badly defined problems and too much focus on fixing bugs versus new product development. Too much bug fix has previously arisen from management prioritizing the construction of too many new features that generates bugs.

"Many issues regarding motivation come from deciding where to put focus, which is a problem that management should solve in their strategy work" - SE

5.4. Company D

"Motivation and process depends on the individual. You need to find a balance. Some want to work very freely, some not. You need to have continuous discussions" - CPO

5.4.1. Administrative controls

The structure of the development department consists of different squads of engineers each led by a squad leader reporting to the Product Director (CPO) and CTO. The roles of engineers have not been formalized and rather tend to navigate around ones core competencies. Nevertheless there are several different formalized processes in the development team according to the CPO. These include processes for development of new features and on boarding of customers.

"It is not possible to have a working development team without a formal development process that follows. It shouldn't be too though, which is something we work a lot with" – CPO

The engineers refers to how their work is structured in two week sprints where each period is ended with a retrospective meeting with the whole squad where not only performance of the engineers are evaluated but also the efficiency of the working routines overall. Consequently, the process is constantly iterated and improved based upon lessons in from these meetings. Additionally, daily stand-ups for 15 minutes exist in each squad and all code are assigned to peer for quality evaluation before it is shipped. There are some formalized coding guidelines that assist engineers in how to write code stylistically and engineers utilize the company's own developed project management tool to update progress of their respective task.

"We evaluate the processes, not only what has been produced [...]. For example, we didn't start with internal code reviews until we started finding small mistakes" – CPO

5.4.2. Planning controls

The company currently has two different products and the goals is to maintain the revenue stream for the old product while gaining revenues of the new product that has just been released.

The long term planning is done by the Chief Revenue Officer, CPO and Technical Lead. The CPO, Technical Lead and Technical lead sits together and divides and prioritizes "EPICS", which is themes based on several user stories that are created by aggregating customer feedback through the customer success department. The prioritized epics and users stories are then presented as a rough sketch on the product roadmap together with time estimates for the next six months but continuously updated each month. This is then presented for the CRO who approves the plan. The squads are constricted to focus on only one large feature at the time.

"We tell people what to do but not how to do it" - CEO

Once prioritization has been done by management features are broken down into tasks together with the engineers in the squad responsible for each feature. The short term planning is done through 2 weeks sprints according to scrum and Kanban principles. Once a feature is completed they have a retrospective to check with everyone that everything works as it is supposed to do. The two weeks sprints does not mean however that specific tasks needs to be done during this period of time. To set time estimates for tasks in these sprints it is up to the person who decides to take the responsibility for a specific task to communicate his estimates to the product director or technical lead and also notify them if problems arise that prolong the process. The engineers also describe that they have very good possibilities to influence what they want to do and not.

"We have retrospectives to understand that the method works rather than the result" - SE

5.4.3. Cybernetic controls

The CTO explains that it is a sense of mastery, purpose and autonomy motivates people and that the use of traditional KPI's removes a lot of autonomy and measures the wrong mastery. Therefore he describes how they use user stories that frames tasks in terms of why, what and who, but leave the how up to engineers. The success of the constructed user story is then measured based upon a few actionable metrics, that can differ depending on the feature, and it is up to the team to draw further conclusions from these to improve further. By using their own project management software they not only keep track of the progress of the individual tasks but also the actionable metrics of the features. Using this, the CPO looks at the output from each sprint on an individual basis to judge the engineers performance.

"I look at everything that has been done under an iteration, per engineer. Based on it, it's quite easy to get a sense if someone produces less than normal" - CPO

The retrospectives are used to measure the development speed in sprints to see if the squads manage to finish tasks on time. However, speed is not considered the most important factor, but rather the quality of the code as it otherwise will require a lot of fixes and will be hard for other people to work with in the system.

"We don't work with time deadlines but quality deadlines" - SE

To ensure quality, code reviews are conducted where usually a more senior engineer evaluates if it reaches up to the coding standards. If not it has to be reworked. According to an engineer this code review also works as a "knowledge sharing instrument to make sure the code always becomes better". Managers also evaluate the number of bugs to get a feeling of the current product quality of their releases through a report comparing fixed bugs vs new unfixed ones. If unfixed are higher then the overall quality level needs to be raised in the squads.

"It's all more based on a gut feeling [...] and I know I have done well if the Product Director tells me so" - SE

Additionally, there are performance reviews two times a year that act as a feedback loop to follow up on how successful engineers are in their role. According to engineers these performance reviews are based on "peer reviews [...] were others in the squad reviews you".

5.4.4. Reward and compensation controls

Engineers receive a competitive base salary and have the possibility to part-take in an option program. However, according to an engineer, the option program "is not considered an incentive as it is a reward [...] it's not the primary driving force". Furthermore, engineers also have a budget to go to conferences to develop their skills further.

"Once people reach a certain income financial incentives do not do as much in terms of motivation, but instead it is important to make people feel a sense of purpose, sense of mastery and a sense of autonomy" - CEO

5.4.5. Cultural controls

The CPO describes that the culture is characterized by "taking personal responsibility and kicking the ball over the line" and the CEO continues by saying that they strive to have an enabling culture where people are told "... what to do, but not how to do it. People have a free will here".

According to one of the engineers the culture is characterized by being "very open [...] its a professional culture but still not formal". He explains that the hierarchy is more flat and that he has more influence here than on his previous jobs.

"My opinion is valued more and people listen to me" - SE

He relates this to the company structure that consists of "squads" that creates a group feeling, and people are motivated by this as there is no one that "commands and decides" on everything that has to be done. The culture between the different development squads are quite alike, and also as the company as a whole, which is because people switch between the different squads and the bi-weekly reviews that takes place gathers the whole company.

5.4.6. Motivation of software engineers

According one of the software engineers you "get kicks when you mange to solve a hard problem and it results in a clear improvement". It is further described that challenging problems are the ones that motivates most. He relates this challenge to his ability to have freedom to decide how to create features.

"We have the freedom to decide ourselves how to create a solution" – SE

The other engineer agrees with this by saying that challenging jobs are the most motivating and that ”setting people up to fail is the best ways for them to succeed”. He further explains how seeing user metrics are motivating as it is nice to know if people actually appreciate the things he builds.

One engineer describes how the squad in itself acts motivating as “you should be able to show the other engineers your code and show your line of thought and motivate why you think this is the best solution”. Therefore, he says, ”you become more strict to yourself and pushed to raise the bar on yourself”.

The same engineer further explains that demotivation is caused if “something takes too long time [...] and you have to scrap something you have spent long time on building”. This happens because the engineer, technical lead and product director have not managed to evaluate it in time to realize how big a project can end up being.

6. Analysis

This section is comprised of an analysis of the motivators of engineers found in all case companies (6.1) and an analysis of how specific configurations of MCS packages in these companies affect these motivators (6.2).

6.1. Motivation of software engineers

As the empirical findings in terms of motivators coincided with Ryan & Deci's (2000) psychological needs of autonomy (6.1.1), competence (6.1.2) and relatedness (6.1.3) the motivators will be grouped under each respective need and then related to the motivators found by Beecham et al (2008). This section will thereby analyse what specific motivators that contributes to satisfying each psychological need for software engineers and how these motivators coincide with the framework of Beecham et al (2008).

6.1.1. Enabling autonomy through proper task definitions, constraints and influence

In accordance with both Beecham et al (2008) and Deci & Ryan (2000), autonomy seems to be a prominent psychological need since what we label as "freedom" is quoted as an important empirical motivator for engineers in all case companies. As software engineers also are autonomous in their nature this is in line with Beecham et al's (2008) findings. Although, compared to Beecham et al's (2008) study were autonomy as a theoretical motivator is reported in only 9 out of 20 studies, its dominance has been showed in the empirical data where all engineers in all case companies refer to the importance of the empirical motivator of freedom and how it is enabled by proper task definitions, constraints and ability to influence as seen in Table 6.

"If you were to limit people in how they work it would get boring" – SE, Company A

"I want interesting tasks and ability to make own choices" – SE, Company B

"We have great freedom in how to solve tasks" – SE, Company C

"We have the freedom to decide ourselves how to create a solution" – SE, Company *D*

Interestingly, there seems to be interrelatedness between the empirical motivator of freedom and how tasks are framed. This is depicted by engineers who explain that freedom is directly related to the ability to decide for themselves how to solve a problem or create a feature,

although it still needs to be specified what is desired and what goal the feature has to be able to practice this freedom. Thus, freedom seems to be best practiced under what we choose to call "proper task definitions" as shown by the below quote by the engineer in Company B. Consequently, this is further highlighted as an engineer in Company C states that badly defined problems can be a source for demotivation. Thereby, one could possibly argue that the empirical motivator of freedom is enabled through "proper task definitions" - not defining "how" the task should be solved but rather "why" and "what" its goal is as stated by the CTO in company D. Connecting this to theory of Beecham et al (2008), the empirical motivator of freedom resembles the theoretical motivator of autonomy as the freedom described by engineers coincides with Beecham's definition of "being able to take own decisions". Moreover, "proper task definitions" partly resembles the theoretical motivator of "identify with task" as it coincides with the definition of "clear goals and knowing the purpose of the task".

"Tell me what you want to achieve and the goal, but don't tell me how to solve it". – SE, Company B

Subsequently, there also seems to be interrelatedness between the empirical motivator of freedom and what we choose to call "balanced constraints" - "frames", "standards" and "structures" for how the development should proceed - as it is said by three developers that they enable this freedom. Consequently, it is shown that freedom is enabled by these constraints as they guide engineers in their continuous work with the tasks at hand. Although, as an engineer in company A states, "limits makes work boring", and an engineer in company B refers to how feeling controlled would be very frustrating, there seems to exist a balance between too much and to little constraints that needs to be maintained in order for it to enable freedom. These "balanced constraints" furthermore resembles the theoretical motivator inherent in SE called "development practices" by Beecham et al (2008).

"[...] You won't grow without freedom, but at the same time you need some frames to guide you" – SE, Company A

"[...] We get full freedom in how to implement stories as long as we follow various standards" – SE, Company C

Additionally, three engineers further emphasizes the importance of being able to influence the direction of the product and making an impact by influencing the design of the plans, what they should build and how tasks are defined. Consequently, what we choose to call the "ability to influence" management appears to enable the empirical motivator of freedom as it provides them with more room to "being able to make own decisions". Subsequently, this "ability to influence" resembles the theoretical motivator of "employee involvement" mentioned by Beecham et al (2008).

Table 6. Summary of motivators satisfying autonomy and its enablers

Psychological need	Satisfied by empirical motivator of	Resembling theoretical motivator of	Enabled by existence of	Resembling theoretical motivators of
Autonomy	Freedom	Autonomy	Proper task definitions	Identify with task
			Balanced constraints	Development practices
			Ability to influence	Employee involvement

6.1.2. Satisfying competence with personal development and complex new work

Interestingly, the quote in section 6.1.1 from the engineer in Company A suggests a connection between the empirical motivator of freedom and personal growth and several other engineers also refers to how the freedom you are given forces you to make own decisions and learn how to solve new challenging tasks. Consequently, it seems as the empirical motivator of freedom also enables what we label as the empirical motivator of "personal development" that was mentioned by seven out of eight engineers and contribute to the satisfaction of the basic need of competence. Thereby, the empirical motivator of freedom and all its enablers thus seem to contribute to the satisfaction of the need of competence as well as autonomy. Connecting it to Beecham et al (2008), the empirical motivator of personal development appears to resemble the theoretical motivator of "development needs addressed" – see table 7 for a summary of all findings.

Additionally, what we choose to call "challenging new work" and "solving complex problems" was further stated as important motivators in themselves for software engineers in the case companies and also resembles motivators that strives towards satisfying the basic psychological need of competence that Deci & Ryan (2000) refer to.

"I get kicks when I solve a hard problem" – SE, Company D

This is found to be in line with the findings of Beecham et al (2008) where the empirical motivator of "challenging new work" resembles "variety of work and the empirical motivator of "solving complex problems" resembles the theoretical motivator of "technically challenging work". One could also say that these two empirical motivators also act as enablers of the empirical motivator of "personal development" as they invariably lead to you growing as an engineer.

I am a problem solver, and if I can't solve problems I get nothing from it [work]" – SE, Company B

Additionally, as mentioned by engineers in Company A and C the empirical motivator of personal development is also enabled by management providing something we call "learning time" - explicit time or money given to engineers in order to learn new techniques on work or at conferences. Looking at Beecham et al (2008), this resembles the motivator "empowerment/responsibility" as software engineers are given the responsibility to develop professionally.

Subsequently, it is also found that what we refer to as "supporting teams" are an empirical motivator in itself as well as an enabler for the empirical motivator of "personal development" as the work in teams both motivates to improve performance due to peer pressure and enables learning. As an engineer in Company D describes, "you become more strict to yourself and pushed to raise the bar on yourself". An engineer in company B shares how the team motivates him as they all have different competencies and specialities which enables them to come with new solutions to his problem and thus enhance his learning. This empirical motivator and enabler resemble Beecham et al's (2008) theoretical motivator of "team work" that is inherent in software engineering.

Conclusively, engineers in three out of four companies describe how "recognition" from colleagues in the team, co-workers in other departments and customer acts as an empirical motivator for them. This in turn could also be directly related to Ryan & Deci's (2000) need of competence as it enables engineers to "feel competent in their encounter with their surroundings". This empirical motivator resembles the theoretical motivator Beecham et al

(2008) refers to as feedback. Important, though, is that this "recognition" is enabled through what we refer to as "structured processes" by created by managers such as daily stand-ups, demo sessions and being provided feedback from customers. Thereby this enabler also resembles the theoretical motivator of "development practices" by Beecham et al (2008).

Table 7. Summary of motivators satisfying competence and its enablers

Psychological need	Satisfied by empirical motivator of	Resembling theoretical motivator of	Enabled by existence of	Resembling theoretical motivators of
Competence	Personal development	Development needs addressed	Freedom	Autonomy
			Challenging new work	Variety of work
			Solving complex problems	Technically challenging problems
			Learning time	Empowerment/responsibility
			Supporting teams	Team working
	Challenging new work	Variety of work	N/A	N/A
	Solving complex problems	Technically challenging problems	N/A	N/A
	Supporting teams	Team working	N/A	N/A
	Recognition	Feedback	Structured processes	Development practices

6.1.3. Creating relatedness through team and recognition

As mentioned in section 6.1.2, both the "supporting teams" and "recognition" can contribute to satisfying the need of competence. But these two empirical motivators can also act motivating through the fulfilment of the third basic psychological need of relatedness (for a summary of our findings see table 8). In essence, what the "supporting teams" and "recognition" can provide coincides with how Deci & Ryan (2000) describes relatedness - "the feeling of being linked to others, part of a group and feel cared for by others" - and how engineers describes the function of their teams and recognition as seen in the below statements. As such, the empirical motivators of "supporting teams" and "recognition" in this case both resembles the theoretical motivator of "sense of belonging" from Beecham et al (2008) and contributes to satisfying the need for relatedness.

"The team. They are very fun people. It's motivating that they come with suggestions and that I can learn from them and they from me. […] We're a bunch of people from KTH that hang out a lot in our spare-time" – SE, Company B

"It's extra fun to deliver something customers see. You really feel that you are making an impact" – SE, Company A

Additionally, as "structured processes" enable the empirical motivator of "recognition" when managers arrange daily stand-ups, demo sessions and interaction or feedback from customers, it appears as the theoretical motivator of "development practices" also enables this empirical motivator when satisfying the need of relatedness.

"It's motivating for engineers to show what they have done and get some applause and recognition" – SE, Company B

Table 8. Summary of motivators satisfying relatedness and its enablers

Psychological need	Satisfied by empirical motivator of	Resembling theoretical motivator of	Enabled by existence of	Resembling theoretical motivators of
Relatedness	Supporting teams	Sense of belonging	N/A	N/A
	Recognition		Structured processes	Development practices

6.2. MCS package configurations affecting fulfilment of psychological needs

From section 6.1 it is shown what empirical motivators that contribute to the satisfaction of the basic psychological needs of autonomy, competence and relatedness for software engineers and what enables their functioning. Consequently, this section will analyse the common configurations of control systems in between all case companies MCS packages that affects these enablers, motivators and thus the satisfaction of the three basic psychological needs. For a summary of the findings of the coming sections please see tables 9 to 11 under each section.

6.2.1. Autonomy satisfied through specific MCS package

As became evident in the analysis of SE motivation in section 6.1.1, the psychological need of autonomy is satisfied by the empirical motivator of freedom that in turn is enabled by the existence of "proper task definitions", "balanced constraints" and "ability to influence". First and foremost, it becomes evident that the "proper task definitions" are made in the action planning stage of *planning controls* used by management in all case companies. Studying their use of *planning controls*, one can draw the conclusion that even though the use of *long range planning* in terms of goals and *action planning* in terms of use of product roadmaps

varies, managers in all companies explicitly emphasize the importance of not defining how to solve a problem or a create features when they decide what to focus development on. One way to further take this into account is to do as the CTO in Company B and give engineers the opportunity to "break down" the tasks and define them themselves.

"In this planning phase, the CPO decides what should be done while the engineers themselves get to decide how it should be done" – CPO, Company C

"We make user stories that describes a what and why. We leave the how to engineers. If this is clear you get great autonomy" – CEO, Company D

Important to note is that this type of action planning would not necessarily on its own enable the empirical motivator of freedom. As found in section 6.1.1, the existence of "balanced constraints" is also an important enabler for this motivator. Therefore, the existence of *administrative controls* in different forms seems to be crucial as they create these standards and frames that can guide engineers when practicing this freedom. For example, in the case of companies A, C and D they all employ *policies and procedures* in the form of coding guidelines that are created together by the team to specify how members should write code and develop according to their standards to ensure that features are built uniformly. Additionally, companies C and D even employ formalized processes for certain steps in development to assist engineers in their work. In the absence of coding guidelines, company B rather employ peer code reviews that ensures that engineers work according to their informal standards. Interestingly, companies C and D also employ these in addition to their other policies and procedures.

One additional aspect of administrative controls that has been found to further enable the empirical motivator of freedom is what we label as the "continuous updates" happening thanks to the *governance structure* employed in the companies with their use of *meetings, deadlines* and *procedures*. Apparent in all case companies is the use of daily stand-ups for the SE teams where each engineer updates others on the status of his task and utilizes his peers to gain any help if needed. This way, engineers can take the help of other engineers and their manager if they need to in order to be assisted in "how" to solve their problem. Moreover, through accessing project management tools such as Kanban-boards or online software programs that structure their work according to their certain procedures, one can also directly

gain information of each developer's status. Additionally, as stated in all companies, these meetings can also be used to communicate adjustment of *deadlines* as soon as engineers encounter a specific problem. This iterative use of deadlines thereby also seems important for enabling the empirical motivator of freedom, as being forced to adhere to a specific deadline would be hard when solving a problem you have not encountered previously.

"I quit my last job since management set unreasonable deadlines that they did not want to adjust. You could not work correctly" – SE, Company B

"When you are developing a new product it is hard to exactly predict how long time everything will take since you encounter new problems along the way" – CEO, Company C

Consequently, these *governance structures* and *policies and procedures* seem to lay the foundation for the *cybernetic controls* utilized by managers where the overall *measures* used to evaluate engineers in all case companies is the quality to their work in relation to the time spent on it. As evident in all companies the evaluation of an engineer is not based on specific numbers, but rather through what we label as a "perception of performance" gathered through the information from *administrative controls* such daily stand-ups and project management tools. An interesting observation comes from the statements made by the CTO's in company C and D pointing to that this informal evaluation seems crucial for the empirical motivator of freedom as a focus on specific measures could affect how engineers choose to solve a problem and thus affect their autonomy negatively.

"Everything you measure can be manipulated. If you start measuring code, engineers will stop helping each other, optimizing the measure and it destroys the team spirit. I rather evaluate their work informally" – CTO, Company C

"As soon as you introduce KPI's you take away autonomy and do the wrong definition of mastery" – CTO, Company D

Moreover, another enabler that is discovered is what the companies and we refer to as "retrospective meetings". This is a *administrative control* that acts as a forum for variance analysis, and thus also as a *cybernetic control*, that enables managers to not only adjust the behaviour of an individual if they have performed badly but also the design of the overall

MCS package used in the company. Since these occur every second week for all companies, either after a sprint as in the case of companies A, B and D, or just as a regular routine for company C, these are used to continually adapt the individual control systems and thus the package of controls based upon the feedback from engineers. Thereby, if engineers were to be dissatisfied of any component, they could voice their opinions to management and the controls could be adjusted. Consequently, one can argue that the retrospective meeting is crucial for SE's to be able to adjust *planning controls* so that tasks are defined in their desired way and thereby enabling the empirical motivator of freedom. Moreover, this enabler can hence be used to modify the MCS package in order to satisfy all psychological needs better.

"Because of our work structure you see really well if one works or not" – CTO, Company B

"We evaluate the processes, not only what has been produced [...]. For example, we didn't start with internal code reviews until we started finding small mistakes" – CPO, Company D

Additionally, as the planning, administrative and cybernetic controls are highly dependent on engineers' ability to communicate with managers and the dissemination of information, it seem as certain cultural controls that we label as "transparency" are another enabler for all other controls to work adequately. The interviewed CTOs maintain that "high ceilings", "open-minded", "transparency" and "enabling" are *values* that permeate the organisations and thus the SEs. Furthermore SE's in company C and D refer explicitly to their ability to voice their opinion as significant of the culture. Thereby, a culture characterized by these values seem necessary to empower engineers to communicate adequately in both planning and administrative controls for them to serve their desired function. Important to note is that, due to the nature of cultural controls, the enabler of "transparency" is evident in affecting all empirical motivators influencing the satisfaction of all three basic needs.

Conversely, the enabler "transparency" also facilitate the third enabler of "ability to influence" as these values encourages engineers to voice their opinions. This ability is also directly dependent on the design of *planning controls* as the decisions about who part-takes in the meetings, how deadlines are set and what information that lays the basis for these plans affect engineers perception of it. As seen in all case companies, managers involve SE in the action planning stage were they can influence the definition of tasks, how they are prioritized and what tasks they want to work with.

Table 9. MCS package enabling satisfaction of the psychological need of autonomy

Psychological need	Empirical motivator	Enabler	Example	Component	Control system
Autonomy	Freedom	Proper task definitions	Not defining how, but why & what	Action planning	Planning
		Ability to influence	Inviting SE to define task, prioritize and set deadlines		
		Balanced constraints	Coding guidelines	Policies and procedures	Administrative
			Formalized processes		
			Peer reviews		
		Continuous updates	Daily stand-ups	Governance structure	
			Adjustment of deadlines		
			Project mgmt tools	Policies and procedures	
		Perception of performance	Informally measure quality and speed through admin controls	Non-financial measures	Cybernetic
		Retrospective meetings	Bi-weekly evaluation of MCS package	Governance structure	Administrative
				Non-financial measures	Cybernetic
		Transparency	Efficient communication between developers and managers	Values	Cultural

6.2.2. MCS package enabling satisfaction of competence

As found in section 6.1.2, the psychological need of competence can be satisfied by the empirical motivators of personal development, challenging new work, solving complex problems, supporting teams and recognition. All these motivators can consequently also be enabled through a certain combination of components in control systems that have been found in the case companies.

As shown in section 6.2.1, there is a specific combination of controls that enable the empirical motivator of freedom. Consequently, as freedom is found to work as an enabler for the empirical motivator of personal development, these controls should also contribute to the satisfaction of the psychological need of competence.

What has been discovered furthermore is that managers enable personal development in some case companies through providing time for engineers to learn new skills or money to go to conferences and gain new insights. Thereby, this use of *reward and compensation controls* seems to enable personal development and thus the satisfaction of the need for competence.

Additionally, the *administrative control* of meetings act as an enabler that we refer to as "feedback" for the motivator of personal development as the daily stand-ups and retrospectives provides forums where feedback are given to engineers that enables them to learn and solve their problems by tapping into the knowledge of the team. This not only facilitates learning for each individual engineer but also for the entire team as lessons get spread throughout the teams when shared in these meetings. This is thus also further enabled by the *cultural controls* existing in the teams described in section 6.2.1.

The empirical motivators of challenging new work & solving complex problems is further found to be enabled by what we choose to call "loose roles" as managers in all companies through *administrative controls* have very broad roles for SE's in their organisational structure as they are based on competence and interests. This in turn enables them to switch in between different fields and thereby encounter challenging new work and complex problems. The informal design of *cybernetic controls* specified by the enabler "perception of performance" in section 6.2.1 subsequently enables this to occur without creating any problems with evaluation. Moreover, the enablers "continuous updates" and "transparency" mentioned in the same section further enables managers to give engineers these challenging new tasks and complex problems, as they trust that they will be able to handle these new problems or speak up if they have any difficulties so they can take help from the rest of the team.

"We don't set real goals. Engineers do the best they can and set an estimate on how fast they can work. I trust that they do their best." – CTO, Company C

Moreover, as "supporting teams" acts as an empirical motivator for engineers to perform better and learn more, it appears as if the *administrative control* of *organisational structure* enables through what we refer to as "peer interaction". This "peer interaction" thus seems crucial for the ability to tap into each others knowledge and create better results as SE's then interact informally on a day-to-day basis. Additionally, the cultural control of *clan controls*

furthermore seems to enable this learning process within the teams through an enabler we refer to as "sub-cultures" as engineers in company A and B refer to how the culture is different within the development department compared to other parts of the company. In company C even the CTO refers to how the culture within IT differs and describes values that emphasize "competence, transparency, open-mindedness and efficiency", which in turn exemplifies how the values within these "clans" could create an environment that triggers this will to perform and develop even more.

As also stated by engineers, gaining recognition served as an empirical motivator that also enhances the satisfaction of competence. This recognition is in turn enabled through what we call "feedback" with the use of *administrative controls* in company B and C where "show and shares" were arranged together with the whole company. Additionally, as customer feedback also provided recognition, the use of *cybernetic controls* to receive feedback on features from users facilitates this. Additionally, this could be used more involving engineers to a larger extent as engineers in company C explicitly state that they would like this.

"More customer contact would be nice. It's fun to see more of the reality than just sitting at your desk and finishing notes and tasks" – SE, Company C

Table 10. MCS package enabling satisfaction of the psychological need of competence

Psychological need	Empirical motivator	Enabler	Example	Component	Control system
Competence	Personal development	Freedom	All examples enabling empirical motivator of freedom	All components in table 9.	MCS package consisting of control systems in table 9.
		Learning time	20% own project time or money for conferences		Reward
		Feedback	Daily stand-ups	Governance structure	Administrative
			Retrospectives		
	Challenging new work & solving complex problems	Loose roles	Roles defined by competence and interests	Organisational structure	Administrative
		Perception of performance	Informally measure quality and speed through admin controls	Non-financial measures	Cybernetic
		Continuous updates	Daily stand-ups	Governance structure	Administrative
			Adjustment of deadlines		
			Project mgmt tools	Policies and procedures	
		Transparency	Efficient communication between developers and managers	Values	Cultural
	Supporting teams	Peer interaction	Peers in teams push and teach each other	Organisational structure	Administrative
		Sub-cultures	Engineering teams have own culture spurring performance	Clans	Cultural
	Recognition	Feedback	Show and shares	Procedures	Administrative
			Customer feedback	Non-financial measures	Cybernetic

6.2.3. Relatedness satisfied through a combination of controls

As found in section 6.1.3 relatedness is satisfied by the same empirical motivators of "supporting teams" and "recognition" that contribute to satisfy the need of competence. Thereby the MCS package that enables the satisfaction of relatedness in this case is the same as that of the empirical motivators of "supporting teams" and "recognition" in section 6.2.2. For example, the use of organisational structure where engineers work within teams and the procedure of show and shares are all dependant on the participation of all team members.

Furthermore, one could possibly argue that relatedness is created through the continuous usage of all components of the MCS packages satisfying the other two needs as they all depend on joint interaction and thus providing "the feeling of being linked to others, part of a group and feel cared for by others".

Table 11. MCS package enabling satisfaction of the psychological need of relatedness

Psychological need	Empirical motivator	Enabler	Example	Component	Control system
Relatedness	Supporting teams	Peer interaction	Peers in teams push and teach each other	Organisational structure	Administrative
		Sub-cultures	Engineering teams have own culture spurring performance	Clans	Cultural
	Recognition	Feedback	Show and shares	Procedures	Administrative
			Customer feedback	Non-financial measures	Cybernetic

7. Conclusions

The purpose of this thesis is to address the academic gap of how the use of MCS affects the motivation of software engineers in technology start-ups. Therefore, this section provides the answer to the question of:

How does the design of the MCS package affect the motivation (self-determination) of software engineers in the context of Swedish tech start-ups?

We will answer this question by addressing the two identified topics outlined in section 1.2. Section 7.1 will address what factors motivate software engineers in technology start-ups and section 7.2 will then address how the MCS packages used by technology start-ups affect these factors found in section 7.1. Thus, answering section 7.1 is a requirement for us to answer the formal research question.

7.1. What motivates software engineers

First of all, our analysis makes it evident that there are certain empirical motivators that have been emphasized as important for the software engineers in the case companies. These empirical motivators are shown to be enabled by what we refer to as "enablers". Furthermore, both these empirical motivators and "enablers" resemble some of the theoretical motivators described by Beecham et al (2008) and are thus classified accordingly. Moreover, these empirical motivators have been related to what psychological need they contribute to satisfying. Hence, in this specific context the satisfaction of each psychological relate to specific empirical motivators that in turn are created by certain "enablers". The specific relationships between needs, empirical motivators, enablers and theoretical motivators can be found in table 12.

Table 12. Empirical motivators satisfying the three basic psychological needs and its enablers

Outcome	Psychological need	Satisfied by empirical motivator of	Resembling theoretical motivator of	Enabled by existence of	Resembling theoretical motivators of
Self-determination	Autonomy	Freedom	Autonomy	Proper task definitions	Identify with task
				Balanced constraints	Development practices
				Ability to influence	Employee involvement
	Competence	Personal development	Development needs addressed	Freedom	Autonomy
				Challenging new work	Variety of work
				Solving complex problems	Technically challenging problems
				Learning time	Empowerment/responsibility
				Supporting teams	Team working
		Challenging new work	Variety of work	N/A	N/A
		Solving complex problems	Technically challenging problems	N/A	N/A
		Supporting teams	Team working	N/A	N/A
		Recognition	Feedback	Structured processes	Development practices
	Relatedness	Supporting teams	Sense of belonging	N/A	N/A
		Recognition		Structured processes	Development practices

Interestingly, the satisfaction of each need is sometimes facilitated by a combination of several empirical motivators. The analysis sheds light upon the satisfaction of the basic psychological needs being dependent on a complex interaction of empirical motivators that in turn are facilitated by "enablers" – certain features that contribute to the existence of the empirical motivators. Malmi & Brown (2008) emphasize the importance of looking at MCS as a package. Likewise is seems as if self-determination can be looked upon as consisting of a package of motivators. Additionally, this classification of Beecham et al's (2008) framework sheds new light on how its described motivators can interact by enabling each other and being dependent on one another by forming a package of motivators.

7.2. How MCS packages affect motivation of software engineers

Analysing the four case companies makes it evident that SE's empirical motivators are affected by certain enablers that relate to specific and sometimes several different MCS components and that these together affect the satisfaction of the three basic psychological needs of autonomy, competence and relatedness. It is also shown that an enabler can consist of several different components in different control systems and that an empirical motivator in turn can be dependent on several different types of components in different control systems. Thereby it is shown that a certain configuration of MCS package is necessary for these empirical motivators to be created and function. Thus, what is found is that affecting these three basic psychological needs are not simply done through the instalment of a certain component in an MCS package, but rather through the existence of a combination of these very specific components within different control systems that together form a MCS package. These packages, which have also been depicted separately in table 9-11, are summarized in their entirety in Appendix 7., due to their vast size.

What is additionally found is that retrospective meetings – a specific component of administrative controls – act as a feedback loop for the whole MCS package, where demands and opinions of engineers on how specific controls should be adjusted is taken into account and acted upon. Thus, the MCS packages in the case companies seem to be adapted continually to the desires of engineers, rather than enforced on them by management, and constantly evolving based upon their needs and opinions. Thereby, one could argue that the three basic psychological needs in themselves could affect how the MCS package should be designed as engineers themselves get a large degree of influence on its design and possibly then could strive to design it to satisfy the three psychological needs.

Lastly, the analysis also shows that a specific MCS package affecting one basic psychological need can affect the satisfaction of several, as for example the MCS package enabling autonomy directly affect the empirical motivators of personal development and thus the satisfaction of the psychological need of competence. Additionally, it is seen that the satisfaction of the psychological need of relatedness is affected by two of the empirical motivators that also satisfy the need of competence and thereby consists of the same MCS components. Conclusively, it is also surprising to see that ordinary reward and compensation controls played a small role in contributing to the satisfaction of any of the needs as their definition is to "focus on motivating and increasing performance of individuals and groups

within organisations", and thus could be assumed to play a central part in motivating employees.

8. Discussion of contributions, limitations and further research

This chapter concludes the thesis by outlining the main contributions (8.1), discussing practical implications (8.2), stating limitations (8.3) and raising areas of further research (8.4).

8.1. Main contributions

Firstly, this study adds to the area of MCS as a package by concluding that it affects the self-determination of SEs through a specific configuration of controls that exist systematically in the researched context. By designing a MCS package that enable self-determination of SEs one can perhaps increase individual performance of SE; and as SE are key in tech start-ups, as shown in the pre-study, our findings is a first step towards gaining a better understanding of how organizational performance could be increased through MCS as a package.

Secondly, we shed light on how components within a control package relates to each other and that certain control systems, such as rewards and compensation controls does not play a central role within the MCS package, as identified by Malmi & Brown (2008). This could seem surprising, as its definition is to "focus on motivating and increasing performance of individuals and groups within organisations". Our research rather shows that its existence serves more as a hygiene factor for SEs to do their work but not to excel at it.

To add to this, our study shows that different components of MCS weigh differently in terms of importance in this particular context. While components such as budgets do nothing to motivate SEs, cultural controls such as the ones in the engineering team plays an essential part. This implies that though Malmi & Brown (2008) portrays all controls of equal worth, they are in fact not.

A major contribution to motivational theory is how the basic psychological needs act as a demand on the design of the MCS package. Through the use of retrospectives, and thus the administrative control component, the design of the MCS package is altered to match the needs of the SEs. This adds to the research of Strauss et al. (2012), suggesting that the administrative control partly acts as an assessment for managers to incorporate the needs of individuals. Its use resembles a finding by Sandelin (2008), who explains that functional demands influence changes in the control packages. Consequently, the MCS package in these

cases appears to be changed continuously due to the functional demands of engineers that revolve around satisfying their three basic psychological needs.

8.2. Practical implications

This paper helps managers understand the crucial question of how individuals are motivated at work. We add to the findings of Chen et al (2014) and conclude that both informal and formal control systems affect motivation. As our findings show that these control systems act simultaneously and are dependent on one another, managers need to consider formal and informal as well as their interdependence when wanting to affect the motivation of individuals.

8.3. Limitations

As the study is of an exploratory form, it makes the findings more descriptive in nature. As a result, the conclusions drawn from this study may lay a base for generalizability, but cannot be confirmed by this study alone. Moreover, more interviews, as well as making a longitudinal study could have made the findings more reliable, by collecting more in-depth data and revealing additional dynamics of MCS and motivation. Additionally, a cross-cultural study could have been made to increase the transferability of the results.

Further on, one needs to take into account the theory of equifinality stated by Sandelin (2008) suggesting that various designs of the MCS package can lead to the same final state. Therefore, other configurations' of the MCS package might yield the same results.

8.4. Areas of further research

The exploratory nature of this study opens up for several areas of future research. For example, one could research more in-depth how these MCS packages are created, what managerial actions that enables them and if social controls (Collier, 2005) plays a role in affecting motivation. It would also be of interest to see how the design of the MCS package differs between different departments within a company i.e. the sales department usually have clear reward controls in terms of incentives. Additionally, it could be studied if there is any specific balance that needs to be maintained in the relationships between the controls as suggested by Mundy (2010). Furthermore, this study could be replicated at a larger sample or other types of companies to evaluate its generalizability. Lastly, a quantitative approach could also be taken to gain more specific results regarding how specific control packages affect motivation.

References

Printed sources

Abercrombie, N. Hill, S. Turner, B, S. 1984. ”Dictionary of Sociology”. Harmondsworth: Penguin

Adams, J.S. Towards an understanding of inequity. 1963. Journal of Abnormal and Social Psychology, 67 (5), pp. 422-436.

Ahrens, T., Mollina, M. 2007. ”Organisational control as cultural practice—A shop floor ethnography of a Sheffield steel mill”. Accounting, Organizations and Society 32 (4–5), 305–331.

Anthony, R. N. 1965. ”Management planning and control systems: a framework for analysis”. Boston: Harvard Business School Press.

Armstrong, A. Davila, and G. Foster. 2010. “Venture-Backed Private Equity Valuation and Financial statement Information,” Review of Accounting Studies, 17/1, pp. 119-154.

S. Beecham, N. Baddoo, T. Hall, H. Robinson, and H. Sharp. 2008. ”Motivation in software engineering: A systematic literature review”. Inform. Softw. Technol., 50(9-10):860–878.

Bell, J. 2006. “Introduktion till forskningsmetodik”, 4th edn, Studentlitteratur.

Baard, P.P., Deci, E.L., Ryan, R.M. 2004. ”Intrinsic need satisfaction: A motivational basis of performance and well-being in two work settings”. Journal of Applied Social Psychology, 34 (10), pp. 2045-2068.

Baumeister, R., & Leary, M. R. 1995. The need to belong: ”Desire for interpersonal attachments as a fundamental human motivation”. Psychological Bulletin, 117, 497-529.

Bedford, David / Malmi, Teemu. 2010. ”Configurations of Control: An exploratory analysis: Working paper”. University of Technology, Sydney.

Braun, V., & Clarke, V. 2006. “Using thematic analysis in psychology. Qualitative Research in Psychology”, 3(2), 77-101.

Bryman, A. and Bell, E. 2007. ”Business research methods”. 2nd ed. Oxford: Oxford University Press.

Carneiro, A. 2000. “How does knowledge management influence innovation and competitiveness?,” Journal of Knowledge Management (4:2), pp. 87–98.

Chen, C.X, Lill, J.B, Vance, T.W. 2014. ”Why Do We Work? Empirical Evidence on Work Motivation and the Effects of Management Control System Design on Work Motivation” AAA 2015 Management Accounting Section (MAS) Meeting

Chenhall, R.H. 2003. ”Management control systems design within its organizational context: Findings from contingency-based research and directions for the future” Accounting, Organizations and Society, 28 (2-3), pp. 127-168.

Collier, P.M., 2005. Entrepreneurial control and the construction of a rel- evant accounting. Management Accounting Research 16, 321–339.

Davila, T. 2005. ”An exploratory study on the emergence of management control systems: Formalizing human resources in small growing firms”. Accounting, Organizations and Society, 30 (3), pp. 223-248.

Davila, A., Foster, G. 2005. ”Management accounting systems adoption decisions: Evidence and performance implications from early-stage/start-up companies”. Accounting Review, 80 (4), pp. 1039-1068.

Davila, A., Foster, G. 2007. ”Management control systems in early-stage start-up companies”. Accounting Review, 82 (4), pp. 907-937.

Davila, A., Foster, G., Li, M. 2009. ”Reasons for management control systems adoption: Insights from product development systems choice by early-stage entrepreneurial companies”. Accounting, Organizations and Society, 34 (3-4), pp. 322-347.

Davila, A., Foster, G., Jia, N. 2010. ”Building sustainable high-growth start-up companies: Management systems as an accelerator”. California Management Review, 52 (3), pp. 79-105

deCharms R. 1968. ”Personal Causation: The Internal Affective Determinants of Behavior”. New York: Academic

Deci, E.L., Ryan, R.M. 2000. ”The "what" and "why" of goal pursuits: Human needs and the self-determination of behavior”. Psychological Inquiry, 11 (4), pp. 227-268.

Deci, E.L., Ryan, R.M. 2000. ”Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being”. American Psychologist, Vol. 55, No. 1, pp. 68-78

Deci, E.L., Ryan, R.M. 1985. ”Intrinsic Motivation and Self-Determination in Human Behaviour”. New York: Plenum Press.

Deci, E.L., Ryan, R.M., Koestner, R. 1999. ”A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation”. Psychological Bulletin, 125 (6), pp. 627-668.

Deci, E.L., Ryan, R.M., Gagné, M., Leone, D.R., Usunov, J., Kornazheva, B.P. 2001. ”Need satisfaction, motivation, and well-being in the work organizations of a former eastern bloc country: A cross-cultural study of self-determination”. Personality and Social Psychology Bulletin, 27 (8), pp. 930-942.

Eisenhardt, K. M., & Graebner, M. E. 2007. Theory building from cases: Opportunities and challenges. Academy of Management Journal, 50 (1), 25-32.

Ryan, R. M., & Deci, E. L. 2008. ”From ego-depletion to vitality: Theory and findings concerning the facilitation of energy available to the self”. Social and Personality Psychology Compass, 2, 702-717.

Demarco, T. and Lister, T. 1999. ”Peopleware - Productive Projects And Teams”. Dorset House.

Eccles, S., Wigfield, A. 2002. ”Motivational Beliefs, Values, and Goals”. Annu. Rev. Psychol. 2002. 53:109–32

Edmondson, A. C., & McManus, S. E. 2007. Methodological fit in organizational field research. Academy of Management Review: In press.

Eisenhardt, K.M. 1989, "Building theories from case study research", Academy of management review, vol. 14, no. 4, pp. 532-550.

Flick, U. 2009, “An Introduction to Qualitative Research”, SAGE Publications Ltd: Fourth Edition

Flamholtz, E. 1996. ”Effective Organizational Control: A framework, Applications and Implications”. European Management JournalVoI. 14, No. 6, pp. 596—611

Flyvbjerg, B. 2006, ”Five Misunderstandings About Case-Study Research.” Qualitative Inquiry, vol. 12, no. 2

Frangos, S.A. 1997. ”Motivated humans for reliable software products”. Microprocessors and Microsystems. 21(10): p. 605-610.

Gagné, M., Deci, E.L. 2005. ”Self-determination theory and work motivation”. Journal of Organizational Behavior, 26 (4), pp. 331-362.

Gagné, M., Zuckerman, M., Koestner, R. 2000. ”Facilitating acceptance of organizational change: The importance of self-determination”. Journal of Applied Social Psychology, 30 (9), pp. 1843-1852.

Ghauri, P., & Gronhaug, K. 2005. “Research Methods in Business Studies”, 3rd Edition. London: Prentice Hall.

Gillham, B. 2005, ”Research Interviewing: the Range of Techniques”, Open University Press, Maidenhead, England.

Grant, A.M. 2007. ”Relational Job Design and the Motivation to Make a Prosocial Difference”. Academy of Management Review, Vol. 32, No. 2, 393–417.

Haustein, E., Luther, R. & Schuster, P. 2014. ”Management control systems in innovation companies: a literature based framework”. Journal of Management Control, vol. 24, pp. 343-382.

Hellmann, and M. Puri. 2002. “Venture Capital and the Professionalization of Start-Up Firms: Empirical Evidence,” Journal of Finance, 57/1 pp. 169-198.

Herzberg, F. 1968. ”One more time: How do you motivate employees?” Harvard Business Review, 46, pp. 53-62.

Huitt, W. 1998, ”Critical Thinking: An Overview”, Educational Psychology Interactive.

Hutzschenreuter, T. 2009. ”Management control in small and medium-size enterprises”. Wiesbaden: Gabler.

Jaworski, B.J. 1988. ”Toward a Theory of Marketing Control: Enviromental Context, Control Types, and Consequences”. Journal of Marketing, Vol. 52 (July 1988)m pp. 23-29.

Kirk, J. & Miller, M. 1986. “Reliability and Validity in Qualitative Research”, Qualitative Research Methods Series 1, Sage Publications, Inc.

Koh, C., Tan, H.S., Tan, K.C., Fang, L., Fong, F.M., Kan, O., Lin Lye, S., (...), Lin Wee, A. 2010. ”Investigating the effect of 3D simulation-based learning on the motivation and performance of engineering students”. Journal of Engineering Education, 99 (3), pp. 237-251.

Langfield-Smith, K. 1997. ”Management control systems and strategy: A critical review”. Accounting, Organizations and Society, 22 (2), pp. 207-232.

Li, Y., Tan, C.-H., Teo, H.-H. and Talib Mattar, A. 2006. ”Motivating open source software developers: influence of transformational and transactional leaderships Proceedings of the 2006 ACM

SIGMIS CPR conference on computer personnel research: Forty four years of computer personnel research: achievements, challenges & the future". Claremont, California, USA 34-43. ACM Press.

Lincoln, Y. Guba, E. 1985, "Naturalistic Inquiry", New Delhi, Sage Publications Ltd

Locke, E.A., Latham, G.P. 1990. "A Theory of Goal Setting and Task Performance". Prentice Hall, Upper Saddle River, NJ

Malmi, T. 2013. " Management control as a package-The need for international research." Journal of Management Control23 (4), pp. 229-231

Malmi, T., Brown, D.A. 2008. "Management control systems as a package-Opportunities, challenges and research directions". Management Accounting Research, 19 (4), pp. 287-300

Maslow, A.H.1970. "Motivation and personality." Harper & Row, New York, USA

Merchant, K.A., & Otley, D.T. 2007. "A Review of the Literature on Control and accountability", (Champman,C. S., Hopwood, A. & Shields, M. D., eds). Handbook of Management Accounting Research 2, Elsevier, Amsterdam, The Netherlands, 785–803.

Merchant, K.A., Van der Stede, W.A. 2007. "Management control system – Performance measurement, evaluation and incentives". 2nd ed., Essex: Prentice Hall

Merchant, Kenneth A & Wim A Van der Stede (2012). "Management Control Systems: Performance Measurement, Evaluation, and Incentives", Prentice Hall, Third Edition

Mitchell, T.R. 1982. "Motivation: new direction for theory and research", Academy of Management Review, Vol. 7, January, pp. 80-8.

Moores, K., and S. Yuen. 2001. "Management accounting systems and organizational configuration: A life-cycle perspective". Accounting, Organizations and Society 26: 351–389.

Mullins, L.J. 2005. "Management and Organizational Behaviour". London: Financial Times. 9th edition.

Otley, D., 1980. "The contingency theory of management accounting: achievement and prognosis". Accounting, Organizations and Society 5 (4), 413–428.

Otley, D., Berry, A. J. 1980. "Control, organization and accounting", Accounting Organizations and Society, Vol. 5, No. 2, 231-244

Patel, R. and Davidson, B. 2011, "Forskningsmetodikens Grunder: att Planera, Genomföra och rapportera en Undersökning", Studentlitteratur.

N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson. 2014. "Software development in start-up companies: A systematic mapping study," Inf. Softw. Technol.,

Reid, G. C. & Smith, J. A. 2000. "What Makes a New Business Start-Up Successful?" Small Business Economics, (14): 165.

Rich, B., Lepine, J. and Crawford, E. 2010. "Job engagement: antecedents and effects on job performance". Academy of Management Journal, 53: 3, 617–635.

Roberts, J., Hann, I. and Slaughter, S. 2004. ”Understanding the motivations, participation and performance of Open Source Software developers: a longitudinal study of the Apache projects”. Carnegie Mellon University Working Paper

Ryan, R. M., & Connell, J. P. 1989. ”Perceived locus of causality and internalization: examining reasons for acting in two domains”. Journal of Personality and Social Psychology, 57, 749–761.

Sandelin, M. 2008. ”Operation of management control practices as a package-A case study on control system variety in a growth firm context”. Management Accounting Research, 19 (4), pp. 324-343.

Sandino, T. 2007. ”Introducing the first management control systems: Evidence from the retail sector ”. Accounting Review, 82 (1), pp. 265-293.

Simons, R. 1995. ”Levers of Control”. Harvard University Press, Boston.

Silverman, D. 2013, “Doing qualitative research: A practical handbook”, SAGE Publications Limited.

Sheldon, Kennon M.; Ryan, Richard M.; Rawsthorne, Laird J.; Ilardi, Barbara. “Trait self and true self: Cross-role variation in the Big-Five personality traits and its relations with psychological authenticity and subjective well-being”. Journal of Personality and Social Psychology, Vol 73(6), Dec 1997, 1380-1393.

Trochim, W.M. & Donnelly, J.P. 2008, ”Research methods knowledge base”, Atomic Dog/Cengage Learning Mason, OH.

Trost, J. 2010, ”Kvalitativa Intervjuer”; Studentlitteratur

Van Lehn, K. 1989. Problem solving and cognitive skill acquisition. In M. I. Posner (Ed.), Foundations of cognitive science (pp. 527-579). Cambridge: The MIT Press.

Vroom, V. H. 1964. ”Work and motivation”. New York: Wiley.

von der Trenck, Aliona; Neben, Tillmann; Heinzl, Armi. 2014. ”The Impact of Self-Determination on the Information-Stopping Behavior of Professionals: An Exploratory Study in the Software Industry”. 25th Australasian Conference on Information Systems 8th -10th Dec 2014, Auckland, New Zealand

White RH. 1959. ”Motivation reconsidered: the concept of competence”. Psychol. Rev. 66:297–333

Williams, G. C., Cox, E. M., Hedberg, V.,, & Deci, E. L. 2000. ”Extrinsic life goals and health risk behaviors in adolescents”. Journal of Applied Social Psychology, 30, 1756-1771.

Williams, G. C., & Deci, E. L. 2001. ”Activating patients for smoking cessation through physician autonomy support”. Medical Care, 39, 813-823.

Witt, L. A., and Burke, L. A. 2002. “Selecting high-performing information technology professionals,” Journal of Organizational and End User Computing (JOEUC) (14:4), pp. 37–50.

Yin, R. 2003. “Case study research: Design and methods", Sage Publications, Inc, vol. 5, pp.11.

Yin, R. 2010. “Qualitative Research from Start to Finish”, Guilford Press, New York.

Zimmerman, M. A. & Zeitz, G. J. 2002. ”Beyond Survival: Achieving New Venture Growth By Building Legitimacy”. Academy of Management Review, 27(3), 414- 432.

Zott, C. & Huy, Q.N. 2007. ”How entrepreneurs use symbolic management to acquire resources”. Administrative Science Quarterly, 52: 70-105.

Other Sources

Carlsson-Wall (2015), Course in “Management Control” at SSE, Dept. of Accounting, Fall 2015.

Knowledge at Wharton, (2015). How Stockholm became a ’Unicorn factory’. Available at: < http://knowledge.wharton.upenn.edu/article/how-stockholm-became-a-unicorn-factory/> [Accessed 11 Mar, 2016].

Swedish Venture Capital Association, (2016). Medlemmar. Available at: < http://www.svca.se/medlemmar/> [Accessed 12 Jan, 2016].

The Telegraph, (2015). How Sweden became the start-up capital of Europe. Available at: <http://www.telegraph.co.uk/finance/newsbysector/mediatechnologyandtelecoms/11689464/How-Sweden-became-the-start-up-capital-of-Europe.html> [Accessed 12 Mar, 2016].

Appendices

Appendix 1. Pre-study interviews

Date	Interviewee	Position, Company	Area of Expertise	Location, Duration
02-02-2016	Dan Skatov	Head of Development	Software Development	Starcounter, 90 min
18-02-2016	Ulf Lewander	Investment Manager, SEB Venture Capital	Tech Industry	Skype, 30 min
15-02-2016	Fredrik Cassel	General Partner, Creandum	Tech Industry	Creandum, 30 min
15-02-2016	Anders Malmgren	Software Developer Consultant, Netlight Consulting	Software Development	Skype, 45 min
08-02-2016	Erik Wegnelius	CEO, Reciva	IT Project Management	Reciva, 60 min

Appendix 2. Different types of control and its components of management control systems

Elements	Components	Description
Planning	Long Range Planning	Goals and actions for the medium and long run are established: has more of a strategic focus.
	Action Planning	Goals and actions for the immediate future, usually a 12-monh period are established and has a tactical focus.
Cybernetic	Budgets	Performance planning and ex post evaluations in relation to the set plan.
	Financial Measurement Systems	Measures that enable a quantification of an underlying phenomena, activity or system.
	Non Financial Measurement Systems	Management initiatives such as TQM.
	Hybrid Measurement Systems	Combines financial measures and non-financial measures such as management by objectives (MBO).
Rewards/ compensation		Attaching rewards and or compensation to achievement of goals.
Administrative	Governance Structure	Monitoring of behaviour through e.g. meetings and deadlines. Additionally looks at who employees are made accountable to for their behaviour by viewing management and project teams.
	Organization Structure	Directing employee behaviour i.e. contact and relationships, through the organizational design
	Policies and Procedures	Specifying how tasks or behaviours are to be performed.
Culture	Clans	Subcultures that probes individuals to a set of skills and values through ceremonies and rituals of the clan.
	Values	Belief system that is communicated formally my senior management to provide values and direction for the organization.
	Symbols	Visible expressions such as workspace design and dress codes to develop a specific type of culture.

Appendix 3. Interview sample for the main study

Date	Position	Company	Interview form
2016-03-16	CEO	A	Face-to-face
2016-03-03	CTO	A	Face-to-face
2016-03-03	Software engineer	A	Face-to-face
2016-03-10	Software engineer	A	Face-to-face
2016-02-23	CEO	B	Skype
2016-03-01	CTO	B	Face-to-face
2016-02-23	Software engineer	B	Face-to-face
2016-03-01	Software engineer	B	Face-to-face
2016-02-22	CEO	C	Skype
2016-02-29	CTO	C	Skype
2016-02-22	CPO	C	Skype
2016-03-01	Software engineer	C	Skype
2016-02-29	Software engineer	C	Skype
2016-02-26	CEO	D	Face-to-face
2016-02-26	CPO	D	Face-to-face
2016-02-26	CTO	D	Face-to-face
2016-02-26	Technical Lead	D	Face-to-face
2016-04-25	Software engineer	D	Skype

Appendix 4. Questions posed in the Pre-study

- What are the biggest challenges for tech start-ups?
- How do tech start-ups differ from other start-ups and larger, more established companies?
- What are the most common factors that hinder tech start-ups for fulfilling their goals?
- How are tech start-ups controlled to accomplish strategic goals?

- What are the biggest challenges in controlling software engineers in tech start-ups?
- How do the software engineering department differ from other departments in tech start-ups?
- How is performance evaluated of software engineers in tech start-ups?
- Hur ser ni att start-ups motiverar sina anställda och särskilt utvecklare?
- Is there anything we have missed that we should have discussed?

Appendix 5. Interview questions posed in the main study

General Questions
Explain your position in the company and your key responsibilities? ○ How long have you worked within this area? ○ What responsibilities do you have that are not included in your formal work description?
What are the company's key objectives? Overall strategy?

Planning	
Main question	Look for
How do you work with planning in the organization?	Short term? vs. Long-term? Budget Involvement of lower-level employees? Perception if you would like that the company planned more and evaluated progress more?
How are goals set within the plans?	The next 12 months? Medium and long-run? Motive of these goals? How is plans/goals communicated within the company?

Cybernetic

Main question	Look for
How do your keep track of your progress in the company?	KPIs (financial vs non-financial) Most important KPI Does the KPI differ in the development department? Reason for measuring these How are KPIs measured How is data collected
How do you work with performance targets in the organization?	What levels of performance are you expected to achieve? (Target-setting) How are these performance standards set? How do you compare performance to standards or targets? How is feedback handled? What happens if performance is better/worse than standards?
Would you describe any challenges/areas of improvement to keep track of the company's progress?	Areas that are hard to evaluate with KPIs? How do you solve this? Wanting the company to be more goal and feedback oriented?

Administrative	
Main question	Look for
How would you describe the organizational structure and roles of the employees?	How the organisational structure looks like What roles do you have in the organisation? What guidelines do you have for specific roles? (Formal or informal)
How is acceptable employee behaviour ensured?	Boundaries for acceptable (employee) behaviour How are these communicated? Formally or informally? How do you precicise/know what is/is not allowed? What happens if an employee violates rules? Would you appreciate clearer guidelines and procedures?

Rewards/Compensation

How are employees rewarded in the company?	Formally vs Informally?
How is the incentive system structured?	What is compensation based on, i.e. is it linked to KPIs?
What is compensation based on, i.e. is it linked to KPIs?	

Culture	
Main question	Look For
How do you work with values in the company?	What is the vision of the company? Is it formalized/written down? How is it communicated? What are the core values of this company according to you? How are the values communicated to employees? Activities with employees? Actions of executives? Symbols in office?
How do you think the company works with recruitment, employee retention, learning and individual development?	How do you think the company works with employee retention, learning and individual development? Do you feel that you are developing? How are new employees introduced in the company? How should an ideal employee be?
How would you describe your work both on an operational and strategic level?	How would you describe your work is related to creativity vs standardized procedures? Do you feel that you are involved in developing the strategy? Do you feel that your opinions have an impact?

Routines	
Main question	Look for
How do the routines of the company look like, if any?	What meetings do you have and how often? What are discussed at these? Who gets to speak? Who is important?

Problem questions	
Main question	Look for
How would describe the flaws do you have in current systems/procedures/routines, if any?	Why does these exist? What are you currently concerned about in the company? Any problems that currently exists?

Motivation	
Main question	Look for
How do you motivate employees within the organisation (engineers especially)?	Recognition of performance? Rewards or incentives? Autonomy? Work-life balance?
What motivates you to work?	Feedback? Personal development? Challenges? Making a difference? Variety of work?
What demotivates you?	Too much control? Too little recognition? Bad communication? To high workload?

Appendix 6. Summary of MCS package & motivation in case companies

Company	Administrative	Planning	Cybernetic	Rewards	Culture
A	One team led by CTO. CPO responsible for product. Report every Monday to MGMT. CTO gatekeeper to protect engineers from disruptions. Roles determined by interests and qualifications. Project MGMT tool structures work. Coding guidelines for how to work. Retrospectives every second week to evaluate process. Daily meetings to evaluate progress. Planning meeting every Monday.	Long term goal to become largest player. No long-term plans in development. Rarely have deadlines. Guided by vision. Not following roadmap anymore. Does not affect individual. Short term plans in sprints based upon prioritization list. Engineers choose tasks, how to solve them and voice opinions on prioritization. Retrospectives held to evaluate sprints. Deadlines set informally through communication.	Hard to evaluate individual engineer. Many values of good output. Retrospective good for measuring performance. Dashboard to track development of product. Evaluation through dialouges. CTO evaluates through "gut feeling". Understanding in team of member productivity. Peer reviews of code and work together on tasks. Monthly development talks.	Base salary without stock options. Budget to go to conferences. 20% laboratory time.	Just formalized in company. Sub-culture in development team. Quality, high ceilings, understanding, not working in silos and community. Rest of company no culture of continuous improvement.
B	Teams of three to four. CTO/CPO same. Technical lead filter in ord. Project MGMT tool structure tasks in sprints. Peer reviews to adjust to standards. Collaboration also adjusts to standards. Roles dependent on needs and interests. Retrospective meetings to evaluate processes.	Two three year goals. Product roadmap for one year with larger features. Engineers breaks down tasks by themselves. Short-term planning by CPO/CTO and other depts to set prioritization, sometimes engineers are invited. Technical lead talks with users to priotitize and sets sprints. Engineers choose tasks and how to solve them. Retrospectives evaluates planning process. Problem with too many tasks, interrupts focus.	Quality vs time evaluation. Adherence to deadlines - no punishment if not done. Important to communicate delays. Evaluated in daily stand-ups. Work structure enables evaluation. Peer reviews quality check. Peer reviews of individual – functioning in team and helping others. Retrospective meeting evaluates performance of sprints. CTO evaluation of perception of productivity – communication with team enables this.	Base salary. Recognition and personal development. Celebrations of launched and show and shares. Friday afternoon of for own project. Challenges and fun more important thans salary.	Formalized values of successful, quality oriented, open-minded, revolutionary, exciting/entrepreneurial. Culture in development department: homogenous – only white male swedes – and neardy and coding interest.
C	Team of five. CTO & CPO. Roles based on competence and interest. Choose tasks themselves.	Three year goal broken down into product roadmap. CTO and engineers involved to set deadlines and	Don't set goals. Engineers set estimates and trust that they work their best. Adherence to deadlines and quality	Base salary and stock option program. Positive feedback	Formalized values challenging, insightful and humble. CTO refers

	CPO acts as filter between other depts to decide features – ensures focus. Engineers feel isolated from customers. Kanban process structuring work. Daily meetings for update. Retrospectives for evaluating process. Features demoed by peers. Coding guidelines and formalized processes to enable learning. Do not formalize too much.	rephrase problems. User stories created and structured in project MGMT tool. Stories prioritized by CTO and engineers – give input on how defined. Kanban then structures work. Engineers choose how it should be done. Build what gives most value.	of work. Not punished if too slow – communication important. Morning meetings with Kanban board evaluate progress. Hard to predict time with new product. Easy to evaluate with Kanban and project MGMT tool. Code test by integration with customer. Peer reviews to ensure quality. Measures becomes manipulated. CTO evaluate informally – has feeling for performance by knowledge and interaction.	upon task completion . Recognitio n at show and share meetings every second week.	to competence, transparency , open-mindedness and efficiency – important for their work. Ability to voice opinion. Premium product. Culture created through interaction.
D	Squads led by squad leader. Roles based on competencies. Formalized processes for new product launch. Need balance. Work structured in sprints. Retrospective evaluate process. Improve the process as you work. Standups. Peer evaluation. Coding guidelines. Project MGMT tool to update progress.	Revenue goals for both products. Long-term planning by top MGMT and technical lead prioritizing EPICS based on customer feedback – becomes product roadmap with time estimates for next 6 months. Updated each month. Squads work with one large feature. Do not tell how to do task. Squad breaks down feature into tasks. Short term planning in 2 week sprints & Kanban. Retrospectives to evaluate process. No deadlines for sprints. Engineer sets estimates himself and communicate delays. Ability to influence what to do.	KPI's remove autonomy and measures wrong mastery. Use stories that frame tasks in why, what and who, but not how. Actionable metrics measure performance of feature. Project MGMT keeps track of progress. CPO evaluates output of each engineer in sprints – sees if someone performes less than normal. Retrospectives evaluate speed and quality. No time deadlines, only quality. Peer reviews evaluates quality – works as knowledge sharing also. Also evaluate number of bugs. Evaluation based on gut feeling – said by engineer. Bi-annual peer reviews.	Competitiv e base salary and option program. Budget for conference s. Financial incentives does not work – need sense of prupose, mastery and autonomy.	Taking personal responsibilit y and kicking ball over line. Enabling culture where people are told what to do, but not how to do it. Free will important. Engineer culture – open and professional, but nor formal. Has a lot of influence – opinion is valued. Squads create group feeling. Culture similar in company due to meetings.

Company	Motivators
A	Freedom in roles, both process and how to solve problems. Control by management "get done when get done". Creative task – artists. Limits makes work boring. Freedom provides growth. Need frames to guide you. Team highly motivating – atmosphere of belief in product and improving it. Solving challenges and delivering solution to customers making impact.
B	Recognition from others and feedback from customers. Challenged in work. Develop himself. Need to have autonomy in work – frustrating to be controlled. Freedom in how to create features. Take own initiatives. Ability to choose how to solve problems. He is problem solver. Team motivates as it enables learning and sharing of ideas. Higher purpose is important.
C	Autonomy in how to solve problems. Personal development key – building company, solving more complex problems and learning new techniques. Affect direction of product – communication with CPO. Recognition at weekly demo sessions and feedback from customers. Bad focus problem for motivation.
D	Challenging problems – affected by ability to make own choices. Freedom to create own solutions. Setting up people to fail är best to succeed. Feedback from customers through feedback. Team acts motivating as others push you to raise bar. Demotivation comes from bad time estimates resulting in scrapping feature.

Appendix 7. MCS package satisfying the three basic psychological needs

Psychological need	Empirical motivator	Enabler	Example	Component	Control system
Autonomy	Freedom	Proper task definitions	Not defining how, but why & what	Action planning	Planning
		Ability to influence	Inviting SE to define task, prioritize and set deadlines		
		Balanced constraints	Coding guidelines	Policies and procedures	Administrative
			Formalized processes		
			Peer reviews		
		Continuous updates	Daily stand-ups	Governance structure	
			Adjustment of deadlines		
			Project mgmt tools	Policies and procedures	
		Perception of performance	Informally measure quality and speed through admin controls	Non-financial measures	Cybernetic
		Retrospective meetings	Bi-weekly evaluation of MCS package	Governance structure	Administrative
				Non-financial measures	Cybernetic
		Transparency	Efficient communication between developers and managers	Values	Cultural

Psychological need	Empirical motivator	Enabler	Example	Component	Control system
Competence	Personal development	Freedom	All examples enabling empirical motivator of freedom	All components in table 9.	MCS package consisting of control systems in table 9.
		Learning time	20% own project time or money for conferences		Reward
		Feedback	Daily stand-ups	Governance structure	Administrative
			Retrospectives		
	Challenging new work & solving complex problems	Loose roles	Roles defined by competence and interests	Organisational structure	Administrative
		Perception of performance	Informally measure quality and speed through admin controls	Non-financial measures	Cybernetic
		Continuous updates	Daily stand-ups	Governance structure	Administrative
			Adjustment of deadlines		
			Project mgmt tools	Policies and procedures	
		Transparency	Efficient communication between developers and managers	Values	Cultural
	Supporting teams	Peer interaction	Peers in teams push and teach each other	Organisational structure	Administrative
		Sub-cultures	Engineering teams have own culture spurring performance	Clans	Cultural
	Recognition	Feedback	Show and shares	Procedures	Administrative
			Customer feedback	Non-financial measures	Cybernetic

Psychological need	Empirical motivator	Enabler	Example	Component	Control system
Relatedness	Supporting teams	Peer interaction	Peers in teams push and teach each other	Organisational structure	Administrative
		Sub-cultures	Engineering teams have own culture spurring performance	Clans	Cultural
	Recognition	Feedback	Show and shares	Procedures	Administrative
			Customer feedback	Non-financial measures	Cybernetic