A Closer Look into Monitoring Processes of Self-directed Teams, that Worked with Scrum

Bachelor Thesis

Bernice Hesselink

University of Twente

June 2016

Faculty of Behavioural, Management, and Social Sciences

Psychology

Supervision: dr. R.E. de Vries & M. Wijga MSc.

Abstract

Nowadays organizations make use of self-directed teams, because of the benefits of these teams. Since these teams have no manager anymore, it is important for them to plan, monitor and evaluate their own actions. This study takes a closer look into monitoring components (content monitoring, monitoring the plan and monitoring progress) of self-directed teams, combined with quality of interaction (ignored, accepted, shared and co-constructed). In this multiple case study three self-directed teams were investigated, that worked with a management method called Scrum. Daily meetings and evaluation meetings were videotaped and coded afterwards. Results show that there is a significant difference between the daily meetings and the evaluation meetings, as well in monitoring processes as in the quality of interaction. Daily meetings relied more on monitoring progress and on accepted interaction, whereas evaluation meetings relied more on content monitoring and co-construction. Moreover, having a certified Scrum-master in a team showed more structured meetings paired with more progress monitoring. Although, there was no significant evidence for this difference between teams. This qualitative finding can be an interesting direction for further research.

Introduction

Self-directed teams have benefits for organizations, because self-directed teams show more competence, improved employee stratification, more innovation, and higher team performance in comparison with traditional teams who were directed by a manager (Hoegl & Gemuenden, 2001; Kauffeld, 2006; Rousseau & Aube, 2010). When teams manage themselves, it is important to plan, monitor and evaluate their own and their team-members actions, since there is no manager anymore who makes sure deadlines will be reached (Kauffeld, 2006).

These processes are called regulation processes (Rogat & Linnenbrink-Garcia, 2011). Self-regulation refers to one's own deliberate planning, monitoring and regulation of behavioral and emotional processes (Hadwin & Järvelä, 2011). Many studies have been carried out on self-regulation in educational context and most of these showed a relation between self-regulation and high performance (e.g. Buckner, Mezzacappa, & Beardslee, 2009; Lane et al., 2008; Winne & Nesbit, 2010). In accordance, Rogat & Linnenbrink-Garcia (2011) found that not only self-regulation, but also socially shared regulation had a positive influence on the quality of the end product. Socially shared regulation refers to the group

function of regulation. Hereby, members share not only goals, but also share regulation activities, such as planning, monitoring and regulating behavior and emotional processes, and outcomes (Schoor, Narciss, & Körndle, 2015). In sum, research found that regulation provides positive results on (team) performance. Furthermore, regulation is well connected to self-directed teams, because people in self-directed teams have more responsibility for setting their own goals and planning, monitoring their own progress and making their own decisions (Carson, Tesluk, & Marrone, 2007). This is why regulation processes are important for self-directed teams.

Monitoring, one of the stages in the regulation processes is seen as crucial for teamwork performance because monitoring affects other components of teamwork, namely backup behavior, feedback and coordination (Dickinson & McIntyre, 1997). Marks & Panzer (2004) confirmed this model of Dickinson & McIntyre (1997). They found that monitoring highly correlates with coordination and feedback, and with great team performance. Results from a case study (Moe, Dingsøyr, & Dybå, 2010) also indicated that when team members fail to monitor each other's progress, teams lose resources caused by limited feedback and back-up behavior affecting their performance. Moreover Langfred (2004) found that a lack of monitoring in self-directed teams, in combination with high individual autonomy and trust, results in lower team performance. "The more individual autonomy there is in a team, the more team members will be working independently of one another, and the more monitoring and communication will be necessary to avoid potential coordination and process losses" (Orton, Weick, Orton, & Weick, 1990 in Langfred, 2004 p. 387). Thus, monitoring is a crucial aspect of regulatory processes for high team performance. Therefore this study focuses on the monitoring aspects in self-directed teams.

Monitoring defined

Monitoring includes checking whether the content is understood during the task and checking the skills that are needed to successfully reach the goal (Santrock, 2011). Monitoring can also refer towards explicit task requirements, task goals and the selected time for the task (Rogat & Linnenbrink-Garcia, 2011). An example of monitoring is to ask yourself "Do I get it?" during the task and wonder what you should do to make it in time (e.g. asking somebody for help) (Santrock, 2011). Monitoring in a team setting refers to the surveillance and awareness from team members of other members' activities (Langfred, 2004). Dickinson & McIntyre (1997) further defined this surveillance and awareness by claiming that this is the recognition when members perform correctly, provide feedback, and back up.

Rogat and Linnenbrink-Garcia (2011) divided team monitoring in three categories; content monitoring, monitoring the plan, and monitoring progress. Content monitoring focuses on checking the team's conceptual understanding of their work on the task, as well the accuracy and quality of the content on the task. A key role in content monitoring is ensuring that team members understand the content, which is accompanied by problem solving (Rogat & Linnenbrink-Garcia, 2011). Monitoring the plan is about the monitoring of the planning by looking at the begin plan and whether the plan needs to be revisited. When a plan is revisited, new plans need to be made. Monitoring progress occurs when teams identify accomplishments, recognize what remained to be completed, and how much time is remaining along the designated plan (Rogat & Linnenbrink-Garcia, 2011). Therefore, in this study, the definition of monitoring reads 'Checking comprehension of the content or explaining the content, checking of the planning is still working, and assessing progress of the task. Comparing a current state with a desired state (goal standard)'.

The quality of socially shared regulation processes

As described above, it is important to make use of socially shared regulation processes (especially monitoring) for self-directed teams, in order to deliver high team performance. Studies (Molenaar, Sleegers, & van Boxtel, 2014; Webb, 2009; Weinberger & Fischer, 2006) found that the effectiveness of collaboration depends on the quality of interaction among the team members. Based on the fact that quality of interaction influences collaboration, Molenaar, Sleegers, & van Boxtel (2014) distinguished different types of intra-group metacognitive (planning, monitoring and evaluation) interaction; namely ignored, accepted, shared and co-constructed activities. Ignored interaction occurs when a team member attempts to control or monitor the team's learning activities, but the other team members ignore this effort. Accepted interaction happens "when team members show their agreement with a metacognitive remark by implementing it in a cognitive activity" (Molenaar et al., 2014 p.313). Shared interaction is defined by Molenaar et al. (2014) as the sharing of metacognitive ideas. Members respond to each other's contributions but they do not build on each other's ideas towards a new idea. When exchanging metacognitive comments result in new ideas, Molenaar et al. (2014) called this interaction co-construction. High quality interaction (sharing and co-construction) enhances learning (Molenaar et al., 2014).

So studies in the educational setting show that shared and co-constructed interaction results in more collaborative learning than when members have ignored or shared interaction (Molenaar et al., 2014; Webb, 2009; Weinberger & Fischer, 2006). When looking at

interaction in organizational teams, there was also found that sharing and co-construction are fundamental processes for team learning (Decuyper, Dochy, & Van den Bossche, 2010). It is important to note that Decuyper et al. (2010) use other definitions than Molenaar et al. (2014) to explain sharing and co-construction. “Sharing is the process of communication knowledge, competencies, opinions or creative thoughts of one team member to other team members” (Decuyper et al., 2010 p. 116). “Co-construction is the mutual process of developing shared knowledge and building shared meaning by refining, building on, or modifying an original offer in some way” (Van den Bossche, Gijselaers, Segers, & Kirschner, 2006 p. 495). The difference in co-constructing and sharing is that with co-construction, sharing takes place in a manner to change traditional thoughts and worldviews, and is seen as the ultimate way for learning (Decuyper et al., 2010). Increased sharing and co-constructing in a team, results in higher performance (Decuyper et al., 2010). So it is recommended for teams to share and co-construct rather than ignore or just accept a standpoint to realize the highest achievable performance. This study used the model of Molenaar et al. (2014) and slightly adapted it with the co-construction definition of Van den Bossche et al. (2006) because of the fact that Van den Bossche et al. (2006) and this study both are about the organizational context and not the school content.

The present study

In this study, the self-directed teams worked with an agile management method called Scrum. Self-directed teams who work with Scrum are small teams, where each team-member has a well-defined role, and the whole team is focusing on a single goal (Rising & Janoff, 2000). Additionally, the working of Scrum facilitates the team to reach this common goal. First, there is a planning phase in which the construction of the project will be discussed, followed by ‘sprints’. A sprint lasts one to four weeks and it produces an observable, usable, deliverable product that implements one or more user interactions with the system. The key idea behind each sprint is to deliver an appreciated amount of work distract from the overall deadline product (Rising & Janoff, 2000).

There are various meetings during the sprint: stand-up meetings, a review meeting and a retrospective meeting (Pfahl, 2014). The stand-up meetings are daily and take up to a maximum of 15 minutes. In these stand-up meetings team-members give answer to three questions: ‘What did you do yesterday?’, ‘What will you do today?’ and ‘Are there impediments?’. During the sprint there are refinement sessions, if needed, which include detailed requirements analysis, splitting large items into smaller ones, estimation of new

items, and re-estimation of existing items (Deemer, Benefield, Larman, & Vodde, 2010). At the end of the sprint there is a review meeting in which the product owner assesses the product. After this, the team evaluates the sprint during a retrospective. Both the review and the retrospective take about one hour (Pfahl, 2014). In this study only the stand-ups and retrospectives were analyzed.

There are three functions fulfilled during the meetings: team members, the Scrum-master and the Product owner (Pfahl, 2014). Team members are self-organizing, work closely together and work in the same formation during one sprint (in a different sprint the formation can change). Every team member is an expert in a particular field but can also take over tasks of other team members. The Scrum-master is responsible for the correct implementation of Scrum and gives support to the team in process-related issues. The Product owner has knowledge about the costumers' needs and describes the requirements towards the team. At the end, the product owner accepts or rejects work results (Pfahl, 2014).

In order to unravel the monitoring processes in self-directed teams, this study aims to answer the following questions:

1. 'To what extent is monitoring used in self-directed teams?'
    1.1. 'To what extent does the type of meeting influence the quantities of monitoring processes?'

The hypothesis is that there is relatively more monitoring during stand-up meetings than in retrospectives, because of the different aims of the meetings. We saw that the stand-up meetings are mainly structured with three questions: 'What did you do yesterday?', 'What will you do today?' and 'Are there impediments?' (Pfahl, 2014) and these questions are in line with monitoring, especially monitoring the progress (identify accomplishments, recognizing what remained to be completed, and how much time is remaining along the designated plan) (Rogat & Linnenbrink-Garcia, 2011). The retrospective is more about evaluation (Pfahl, 2014) and therefore it is expected that there is less monitoring in retrospectives than in stand-ups.

    1.2. 'What are differences between teams regarding monitoring processes?'
2. 'What is the quality of the used monitoring processes in terms of interaction (ignore, accepting, sharing, and co-construction)?'
        2.1 'To what extent do monitoring processes (monitoring the content, plan or progress) interact with the quality of interaction (ignored, accepted, shared or co-constructed)?'

The hypothesis is that monitoring the content relies more on shared or co-constructed interaction than on ignored or accepted interaction, because this form of monitoring often

goes along with collaborative problem solving followed by feedback and clarification (Rogat & Linnenbrink-Garcia, 2011). Feedback and clarification are high quality interaction processes (Molenaar et al., 2014).

Method

Design & Participants

The design of this study was a multiple case study, consisted of three scrum teams of the same organisation, working in the software context. The organisation is located in The Netherlands and delivers high quality digital services with a focus on the digital government. The employees are freelancers. There were 14 participants in total (n = 14).

Team 1 consisted of four members, three men and one woman, three a had higher education degree and one had a university degree. The average age of this team was 41 (ranging from 38 to 45). This team had a certified Scrum-master and one product-owner, the other two members filled the role of team-members.

Team 2 consisted of six members (all men), four of them had a higher education degree and two a university degree. The average age of this team was 38 (ranging from 35 to 49). This team had an uncertified Scrum-master, the other five members filled the role of team-members.

Team 3 consisted of four members (all men), whose two had a higher education degree and two had a university degree. The average age of this team was 35 (ranging from 32 to 38). This team had an uncertified Scrum-master, the other three members filled the role of team-members.

In this study 15 meetings were analysed, consisting of three retro-meetings and twelve stand-ups during one sprint. The retro-meetings lasted from 30 minutes to around one hour (in total 127 minutes). The stand-ups varied between 7 and 15 minutes (in total 128 minutes). There were six stand-ups and one retrospective coded of team one. In team two and three, there were three stand-ups and one retrospective coded per team.

Materials & Procedure

All team meetings were observed by means of a 360 degrees video camera. In order to analyse the regulation processes, especially monitoring, the videos were coded using a coding scheme (table 1, 2, 3) described in ‘Analysis’.

The study was presented to each team, where the procedure of the study was explained. During this explanation, the aim of the study, the requested time investment, the method of

the data collection, and the surveillance of privacy were discussed. Afterwards, the team members made the decision whether they wanted to participate or not (the team only participated when all members wanted to). Next, there was a pilot of two weeks where participants got used to the camera. After this period, team members again had the opportunity to withdraw from the study. Furthermore, the researchers signed a confidentiality agreement. The team members signed an informed consent, which emphasized that the data will only be published anonymous and confidential to third parties, participation was voluntary and the team members had the right to terminate their participation at any time without giving reasons. Finally, the team members were videotaped during the meetings when they agreed, the researcher switched on the camera at the beginning of the meeting. Sometimes the meetings of the different teams were scheduled at the same time (so the researcher was not present in one of them), then one of the team-members switched on the camera.

Analysis

In order to analyse to what extent monitoring was applied during the meetings, table 1 was used relying on the definitions of Rogat & Linnenbrink-Garcia (2011). Different regulation processes were coded. Planning and evaluation were just coded to make a distinction of monitoring, but they do not come back in the next steps of the scheme. Non-regulation activities were not coded. Every time team-members started a new regulation process or a new topic, it was coded with a new regulation code. Planning occurred when the teams were making a strategy how to meet the deadline e.g. “We must finish this part tomorrow, so should I come to your desk after this meeting to finish the story?” Monitoring was coded when a team-member was checking their own or colleges’ comprehension, planning and progress e.g. “I have finished the URL’s yesterday, did you see it? And are there any questions about it?” Evaluation occurred when teams where reflecting on the good and bad points e.g. “I am satisfied about the way we fixed the URL’s, we have worked hard”.

Table 1
Regulation processes (Rogat & Linnenbrink-Garcia, 2011)

Code	Description
Planning	Discussing how to go about solving problems, discussing strategies, goal setting, collaboratively discussing task directions, translating directions into a clear plan and designating tasks.

Monitoring	Checking comprehension of the content or explaining the content, checking of the planning is still working, and assessing progress of the task. Comparing a current state with a desired state (goal standard).
Evaluation	Making a judgement about goal attainment.

A second reviewer coded 20% of the total data (20 minutes of a retrospective and 2 stand-ups, distributed between the 3 teams) in order to calculate Cohen's Kappa in this first step of the coding scheme. Kappa resulted in *0.528*. According to Landis & Koch (1977) this kappa value should be considered as a moderate strength of agreement.

In the next step (sub code 2), a closer look to monitoring was taken. The various components of monitoring were analysed by the definitions of Rogat and Linnenbrink-Garcia (2011). Hereby monitoring processes were divided in monitoring the content, monitoring the plan and monitoring the progress (table 2). Content monitoring was about comprehension of the content of the tasks e.g. explanation and justification aimed at fostering conceptual understanding; team members' feedback, coupled with explanation and collaborative problem solving. Monitoring the plan was about restructuring of the original planning e.g. members would return to monitoring the original task plan in order to clarify what they should be working on, revisit what they had proposed as next steps, or adjust the plan as needed. And monitoring progress was about checking what the teams already had accomplished and what they have to do more to meet the goals e.g. recognize what had already been accomplished, followed by identifying the next steps of the task or what remained to be completed.

Table 2
Components of monitoring with examples

Code	Description	Example
Content monitoring	Focused on monitoring content understanding	"So if somethings fails on the dashboard, there is now a second opinion, but it is still completely unclear why they behave differently, because they are completely identical, everything is the same now, just the way how we started them"
Monitoring the plan	Referred to teams revisiting the task directions or revisiting the task or content plan they had set for accomplishing the task	"I found out that the LTC's BFM wrote, were wrote from a wrong perspective '…' So I have to create a new logical test. That means that my other work lapses"
Monitoring progress	Occurred when teams checked whether they were meeting set goals and/or making sufficient progress along the designated plan as well as attending to how much time remained relative to the work remaining	"Yesterday I was started with the functionality and I did come a long way. The webpages, the controls and the outside are finished. I already saw that the ping does not work, but the Story is not finished yet"

Third, there has been analysed what the quality of interaction (ignored, accepted, shared or co-constructed) was. To make a distinction between this interaction qualities, the model of (Molenaar et al., 2014) was used (table 3), combined with the definition of co-construction of Van den Bossche et al. (2006). The accepted interaction code of Molenaar et al. (2014) was also slightly adapted from 'group members show their agreement with a metacognitive remark by implementing it in a cognitive activity' to a more suitable definition of the accepted interaction of monitoring processes: 'When the team members engage in a monitor activity and only accept or deny a statement'. We did this to make a better distinction between accepted and shared interaction, and because of the fact that cognitive activities could not be observed in this study.

Ignored interaction was coded when team members did not react to other team members, and for example began to talk about a new topic. In the example given by the table, there is a switch to another topic and there is no response to 'John' (the names are fictional, due to privacy reasons). Accepted was coded when team members confirmed or denied a statement with a few words like "Yes, beautiful" or "No, that is not what I said, but let's move on". Shared was coded when interaction was more detailed and was more in depth than accepting alone, see table for an example which did go more in-depth than the accepting code. Co-construction could be called the highest level of interaction in this study. Co-construction stood for shared interaction but also for a change in a worldview. As we can see in the example of table 3, the worldview of 'Will' is changing. First he had the idea that product-owners have a lot to say in their team, but by the end he knows that the team itself can decide what to work on.

Table 3
The quality of interaction

Code	Description	Example
Ignored	When the team members do not relate to nor engage in another team member's monitoring activity.	John: " Thus the TLS is working for all profiles and the rapport t is also working. I fully checked it. Now, I am working on the Story of the pretty URLs. I already finished the URLs for the best effort '...' now I am working with LBL message and that is more difficult" Oliver: "Did you [pointing to Rob] put something in the mail about the HDLs yesterday?"
Accepted	When the team members engage in a monitor activity and only accept or deny a statement.	John: "I worked on the certificate validation Friday, to be more precisely I worked on the validation of the signature '…' That is built-in and I think I finished it correctly, now I am hoping to finish the review with Will and then I will start with a new storyboard. That is it for me" Kim: "Ok, super."

Shared	When a team member relates to a monitoring activity share with another member, give feedback, suggestions, and explanations.	Kim: " Today, I will look to the https-profiles, thus the profiles without configuration. I will play around with this profiles and when I got stuck, I am going to call Rob." John: "How far are we with the testing of all profiles. Do you think that we will finish it this sprint?' Kim: "The https-profiles can be finished by the end of this sprint' John: "Ok, but by the end.." Kim: "I need to start first" John: "You do not know yet how long it takes, you first have to start?". Kim: "Yes"
Co-construction	When team members not only relate to but also engage in each other's monitoring activities'. This is more than sharing, members have to change original thoughts and worldviews.	Will: "One clarification from the Scrum-master about the process" John: "Yes you always may" Will: "We are the owners of the sprint backlog. In the past my experience is that developers supposed or were not allowed to put stories on the sprint backlog. Because they can create an enormous amount of work out of nowhere, and there is then no control of the product-owner about the activity which happens in the sprint. So the procedure, what kind of rules, even it is necessary, you have to communicate before. So if you create something just make it and pull. But if the product-owner have other priorities? How does it work?" John: "Very simple, the product owner does not decide what we do in the sprint. In fact she has nothing to say what we do in the sprint. The only thing she can say is I have a list what must be done and this is the priority of the things I want done, and I ask you to do them. And if you think that there are things that need to be quicker done, then it is our responsibility of the team to do them and put them on the product backlog." Will: "So if somebody come to check my story and say it has to be done, because he or she gives a presentation about it then I put all the other work to the side and I finish it." John: "No wait: Do you want to finish it? Actually does our team want to finish it? As we want as a team to finish it then yes". Kim: "No is an option" Will: "So it means that we decide that this is important to have continuous integration, and it takes a lot of time.." John: "Then we will do it."

When a meeting was coded according this codebook above, the output was looking like this (table 4). Every time monitoring was coded, there was also examined which component of monitoring was used (sub code 2) and afterwards the quality of this interaction was identified (sub code 3). In table 4 can be seen that from 07:13 to 8:30 no regulation process was observed, so these minutes were not coded.

Table 4
An example of the coding scheme

Time	Subcode1	Subcode2	Subcode3
00:00:56 - 00:01:48	Monitoring	Progress	Accepted
00:01:49 - 00:01:59	Planning		

00:02:00 - 00:02:27	Monitoring	Progress	Accepted
00:02:28 - 00:04:14	Monitoring	Planning	Shared
00:04:16 - 00:07:13	Planning		
00:08:30 - 00:10:14	Monitoring	Progress	Co-constructed

Results

First, the quantitative results are shown. This starts with the analysis of the descriptive statics (frequencies). Next, the differences between the meetings (stands-ups vs retrospectives) were investigated. During the third step of the quantitative results, there was analysed to what extent the teams differ in the use of monitoring. In the last step of the quantitative results, there was examined to what extent the sub codes (2 and 3) showed a pattern. Finally some qualitative notions about the data will be presented.

Quantitative results

Monitoring was coded 81 times. Looking closer to monitoring, content monitoring was coded 30 times, monitoring the plan 10 times and monitoring the progress 41 times (table 5). Next, looking at the interaction during the monitoring activities, ignored interaction was coded 3 times, 46 times accepted, 24 times sharing and 8 times co-construction (table 6).

Table 5
Frequencies of monitoring components

Code	Frequency	%
Content monitoring	30	37,0
Monitoring the plan	10	12,3
Monitoring progress	41	50,6
Total	81	100, 0

Table 6
Frequencies of the quality of interaction

Code	Frequency	%
Ignored	3	3,7
Accepted	46	56,8
Shared	24	29,6
Co-construction	8	9,9
Total	81	100,0

Table 7 shows the differences between the stand-up meetings and retrospectives. When looking at table 7, there can be seen that stand-up meetings rely more on monitoring than the retrospectives do (representatively 69 and 12). Especially monitoring progress is much more used in stand-ups compared with retrospectives (40 compared with one). When running a chi-square analysis, there was a significant association between the type of meeting and the

components of monitoring χ^2 (2) = 13.31, $p < .00$. The post hoc analysis shows that stand-ups rely more on progress monitoring and retrospectives rely more on content monitoring (table 8).

About the quality of interaction, there can be seen that accepted interaction occurs more frequent in stand-ups than in retrospectives (43 compared with 3, table 7). When preforming a chi-square analysis between the different meetings and the interaction qualities, there was a significant association between the type of meeting and the quality of interaction χ^2 (2) = 6.35, $p = .04$. In this analysis the code of ignored was left out, because of the relative small total amount (3 codes, 3.7%). The post hoc analysis (table 9) show that stand-ups rely relatively more on accepted interaction than retrospectives. In turn, retrospectives rely relatively more on shared and co-constructed interactions.

Table 7
Differences between stand-ups and retrospectives

Meeting	Content monitoring	Monitoring the plan	Monitoring progress	Total	Ignore	Accept	Share	Co-construct	Total
Stand-up	20	9	40	69	2	43	19	5	69
Retrospective	10	1	1	12	1	3	5	3	12
Total	30	10	41	81	3	46	24	8	81

Table 8
Chi-square analysis between the meetings and the components of monitoring

		Content monitoring		Monitoring the plan		Monitoring the progress		Total	
Total	Count	20	**10**	9	**1**	40	**1**	69	**12**
	Expected Count	25.6	**4.4**	8.5	**1.5**	34.9	**6.1**	69.0	**12.0**
	Adjusted Residual	-3.6	**3.6**	0.5	**-0.5**	3.2	**-3.2**		
	Count	30		10		41		81	
	Expected Count	30.0		10.0		41.0		81.0	

Note. The retrospectives are in bold

Table 9
Chi-square analysis between the meetings and the quality of interaction

		Accepted		Shared		Co-constructed		Total	
Total	Count	43	**3**	19	**4**	5	**3**	67	**10**
	Expected Count	40.0	**6.0**	20.0	**3.0**	7.0	**1.0**	67.0	**10.0**
	Adjusted Residual	2.1	**-2.1**	-0.8	**0.8**	-2.2	**2.2**		
	Count	46		23		8		77	
	Expected Count	46.0		23.0		8.0		77.0	

Note. The retrospectives are in bold

The differences between the teams about the various components of monitoring are shown in table 10 and the differences between teams' quality of interaction during monitoring are shown in table 11. The results in table 10 give the impression that team 1 monitored the progress relatively more than team 2 and 3 (the count is higher than the expected count), and team 2 relatively more monitored the content. Although the teams and sub code 2 did not show a significance relation ($p > 0.05$).

In table 11 there can be seen that team 1 interacts relatively more on co-constructed interaction, team 2 on accepted and shared interaction. However, there was no significance relation between the quality of interaction and the teams ($p > 0.05$).

Table 10
Chi-square analysis between teams and the components of monitoring

Team		Content Monitoring	Monitoring the plan	Monitoring the progress	Total
1	Count	11	5	23	39
	Expected Count	14.4	4.8	19.7	39.0
	Adjusted Residual	-1.6	0.1	1.4	
2	Count	10	3	8	21
	Expected Count	7.8	2.6	10.6	21.0
	Adjusted Residual	1,2	0.3	-1.3	
3	Count	9	2	10	21
	Expected Count	7.8	2.6	10.6	21.0
	Adjusted Residual	0.6	-0.5	-0.3	
Total	Count	30	10	41	81
	Expected Count	30,0	10,0	41,0	81,0

Table 11
Chi-square analysis between teams and the quality of interaction

Team		Accepted	Shared	Co-construction	Total
1	Count	22	10	6	38
	Expected Count	22.7	11.4	3.9	38.0
	Adjusted Residual	-0.3	-0.7	1.5	
2	Count	14	5	1	20
	Expected Count	11.9	6.0	2.1	20.0
	Adjusted Residual	1.1	-0.6	-0.9	
3	Count	10	8	1	19
	Expected Count	11.4	5.7	2.0	19.0
	Adjusted Residual	-0.7	1.3	-0.8	
Total	Count	46	23	8	77
	Expected Count	46.0	23.0	8.0	77.0

To analyze if there was a pattern between the various processes of monitoring (content, the plan and progress) and the quality of interaction, a chi-square was carried out. Here, ignored

interaction was also left out, because of the relative small total amount. Looking at table 12, there can be seen that co-construction did occur relatively the most by content monitoring (5 times, were 2.9 was expected) and that monitoring the progress most often went along with accepted interaction (24 times). However, there is no significant evidence for an association between the components of monitoring and the quality of interaction ($p > 0.05$).

Table 12
Interaction between the components of monitoring and the quality of interaction

		Accepted	Shared	Co-construction	Total
Content monitoring	Count	15	8	5	28
	Expected Count	16.7	8.4	2.9	28.0
	Adjusted Residual	-0.8	-0.2	1.6	
Monitoring the plan	Count	7	3	0	10
	Expected Count	6.0	3.0	1.0	10.0
	Adjusted Residual	0.7	0.0	-1.2	
Monitoring the progress	Count	24	12	3	39
	Expected Count	23.3	11.6	4.1	39.0
	Adjusted Residual	0.3	0.2	-0.8	
Total	Count	46	23	8	77
	Expected Count	46.0	23.0	8.0	77.0

Qualitative results

In this section of the results, the qualitative results of the data will be discussed. The aim of this discussion is to give a more detailed description of the differences between the teams.

In team 1, the meetings were very structured and everyone was given the opportunity to speak. Team-members answered the 3 questions ‘What did you do yesterday?, ‘What will you do today?’ and ‘Are there impediments?’ consequently. Here, the Scrum-master made sure that everybody told something about the progress during stand-ups about the sprint. This became evident during the meetings, where all team members spoke in every stand-up meeting. In the retrospective the Scrum-master was sitting among the other members and they again did a structured meeting. They had notes for strong and weak points and discussed them one after the other. The Scrum-master was also open for criticism and wanted to change his behavior in order to do a better job: “Please do me a favor, if I show this kind of behavior, remind me of it, and I will ensure you I change the behavior immediately, because I do recognize it”. This team gave a positive impression during meetings, and the Scrum-master often emphasized these positive points and complimented the team with positive feedback: “I want to add some little things, if we talk about this kind of things as an improvement, than we are working on such little things to improve, that all the big things already have been done. Which means that our team operate very, very, very efficiently”.

Team 2 had less structured meetings than team 1, and sometimes there was a negative atmosphere. During the stand-ups, the 3 monitoring questions were not consistently answered, and besides, some team-members did not speak at all. The retrospective represented the negative atmosphere. This meeting consisted partly of someone who had no motivation anymore. The Scrum-master tried to find a solution for this team-member: "Is there something else you can do so you will get your motivation back?", but the team-members just answered with no, and the other team-members made jokes of it. So the Scrum-master tried to solve the problem, but it seemed like nobody took him seriously.

The stand-up meetings from team 3 were also less structured than the meetings of team 1. This became evident during the stand-up sessions in which not every member actively participated and the three monitoring questions were not used to structure the meeting, as was the case in team 1. The retrospective session, however, appeared to be more structured. The Scrum-master stood in front of the team and wrote down the positive and negative points the team mentioned about the sprint. The Scrum-master clearly acted as a chair of the meeting, making sure every point was discussed.

Conclusion & Discussion

This study aimed to investigate monitoring in self-directed teams, looking further to the components of monitoring and the quality of interaction. In this study, three self-directed teams were analyzed to answer the research questions.

The first conclusion is about the first research question: 'To what extent is monitoring used in self-directed teams?'. When examining the results, there can be said that monitoring is used in self-directed teams. Looking closer, monitoring is more used in stand-ups than in retrospectives.

This is in line with the hypothesis, suggesting that monitoring meets the objectives of the stand-ups (responding to three monitoring questions) more than the objectives of the retrospectives (discussing good and bad points) (Pfahl, 2014). Furthermore, the hypothesis was that stand-ups relied most on monitoring the progress, because of the fact that the answers of the three questions match with monitoring the progress defined by Rogat & Linnenbrink-Garcia (2011). In agreement, the results showed significantly that stand-ups relied the most on monitoring the progress. Thus monitoring is used more in stand-ups than in retrospectives, this suggests that monitoring is needed for stand-ups in order for them to go well. Furthermore, monitoring the progress is the most used component of monitoring during

stand-ups, and this suggests that monitoring the progress is recommended for stand-ups, in order to facilitate the overall work project (Rising & Janoff, 2000), and achieve high team performance.

About question 1.2., 'What are differences between teams regarding monitoring processes?', could be said that there are no statistical differences between the teams, not in relation to the components of monitoring and also not in relation to the quality of interaction. However, the results showed in a certain way that team 1 relatively more monitored the progress than the other teams. This is in line with the impression they gave during the meetings. These meetings were structured, this became evident when every team-member consequently answered the three monitoring questions explained by Pfahl (2014) during the stand-ups. The Scrum-master of team 1 was the reason why these meetings were structured, by ensuring that every team-member did answer the questions. Studies show that structured processes produce higher quality solutions, which result in higher team performance (Rogelberg, Barnes-Farrell, & Lowe, 1992 in Guzzo & Dickson, 1996). Additional, the fact that this team was the only team that had a certified Scrum-master can also make a difference. Research exposed that agile teams who had a trained coach in comparison with an untrained coach score different in productivity gains, in survey comments, and in individual teams' management satisfaction levels (Benefield, 2008). Benefield (2008) recommends certified coaches, because according to her calculations these coaches provide an increased productivity of around 30%, and will save the company 1,5 million dollars a year. Moreover Barnes, Pashby & Gibbons (2002) showed that good project management is essential to success, which can be achieved by a structured setting and good progress monitoring. In conclusion, there was no significant proof for differences between the teams. However, the qualitative results showed that a qualified Scrum-master is required for structured management during the meetings, and this can lead to higher achievements in the end.

The second research question was: 'What is the quality of the used monitoring processes in terms of interaction (ignored, accepted, shared, and co-constructed)?' The results showed a significant association between the different meetings and the quality of interaction. The stand-ups relied relatively more on accepted interaction, while the retrospectives relied more on shared and co-constructed interaction. Question 1.1 'To what extent does the type of meeting influence the quantities of monitoring processes?' is also answered with this. As said in the introduction, it is recommended for teams to share and co-construct in order to achieve high team performance (Decuyper et al., 2010). However in this study, it is striking that shared and co-constructed interaction is less used in stand-ups. Reasons for this can be the

limited time and the aims of a stand-up. In a stand-up, all team-members have to answer What did you do yesterday?', 'What will you do today?' and 'Are there impediments?' (Pfahl, 2014) in about 15 minutes. In these 15 minutes there is not that much time to give each other feedback, discuss or co-construct, and accepted interaction occurs. If there are any uncertainties about the sprint, team-members can discuss this in a refinement (Deemer et al., 2010), so discussion and clarification of the sprint occurs during refinements, and this does not have to take place in stand-ups. Maybe it is more recommended for a stand-up to have mostly accepted interaction and thus fast confirming statements. Molenaar & Chiu (2013) showed already that accepted interaction have positive outcomes for team processes. In agreement, other research showed that accepted interaction provides higher employee satisfaction in agile teams: "One of the main factors associated with enjoyment and excitement in agile software development teams was the ease and speed by which team members could get things done; questions were answered, problems were resolved, and collaborative opportunities were quickly grasped" (Whitworth & Biddle, 2007 p. 64).

The sub question about the second research question was: 'To what extent interact monitoring processes (monitoring the content, plan or progress) with the quality of interaction (ignored, accepted, shared or co-constructed)?'. The hypothesis was that content monitoring relies more on shared and co-constructed interaction. The results show that content monitoring did go relatively more together with shared and co-constructed interaction. Besides, monitoring the plan and progress interact less with these qualities of interaction, because most of the time when a plan needed to be reconsidered or when progress was checked, people discussed less and faster accepted statements. However the results did not show significant proof for this hypothesis.

This study is not without limitations. First, because of limited time, only the first step of the coding scheme was coded by a second reviewer to calculate Cohens' Kappa. To know of the coding scheme is reliable as a whole it needs to be coded as a whole.

There is not enough data to say something about the effects over time, because the data was collected in only 2 weeks. Also, the data is not representative for all teams in organisations. Therefore we need data of more organisations. Moreover, all meetings should be coded, not only the stand-ups and retrospectives but also the refinements and reviews, to make more conclusions of the various meetings. Finally more data, especially the data which shows the final team-performance (e.g. is the deadline reached in a sufficient way) should be included to see to what extent the components of monitoring and the quality of interaction interact with the team performance.

It is also a limitation that there was more data in this study from team one than from the other two teams, because in this study team 1 could and is more adequate and detailed described than the other two teams, what may give a bias of the quantitative results.

Sometimes it was hard to separate the regulation process planning from monitoring the plan. The difference was that monitoring the planning relies on restructuring the planning, which was made before (during the regulation process). But it was not always clear if there already was planning or if they made it by that moment.

Finally, it was sometimes difficult to know when they discussed a new topic, because they were working in the software context and some matching terms are maybe interpreted as a new topic, because a lack of knowledge about soft-ware terms of the researcher.

Further research

In sum, this study underscores interesting results about monitoring in self-directed teams. This study looked at the components of monitoring and at the quality of interaction. First, the teams relied more on monitoring the progress in stand-ups, on the other hand the teams relied more on content monitoring in the retrospectives. It is recommended to investigate the relationship of these components of monitoring, during the various meetings, with final team-performance. Only then, there can be concluded if these components are important for particular meetings which provide team performance.

Besides, the teams relied more on accepted interaction than on shared or co-constructed interaction during the stand-ups. That is why this study recommends to analyze if accepted codes during stand-ups are statistical relatable with final team performance in further research. Only then, there can be concluded to what extent the quality of interaction in particular meetings relates with team performance.

Also the differences between agile teams with a certified Scrum-master in comparison with teams who had an uncertified Scrum-master are attention-grabbing and should be investigated more, because this brings benefits to organizations (Benefield, 2008). It was remarkable that the team with the certified Scrum-master had more structured meetings and relied relatively more on progress monitoring. Thus, these two components (structured meetings and progress monitoring) can be achieved by a trained Scrum-master and result in success for companies (Barnes et al., 2002; Benefield, 2008).

I expect that this study will help advance this line of research by providing a codebook where components of monitoring and the quality of interaction are combined.

References

Barnes, T., Pashby, I. A. N., & Gibbons, A. (2002). Industry Interaction : A Multi-case Evaluation of Collaborative R & D Projects, *20*(3), 272–285.

Benefield, G. (2008). Rolling Out Agile in a Large Enterprise. *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, 461–461. doi:10.1109/HICSS.2008.382

Buckner, J. C., Mezzacappa, E., & Beardslee, W. R. (2009). Self-regulation and its relations to adaptive functioning in low income youths. *The American Journal of Orthopsychiatry*, *79*(1), 19–30. doi:10.1037/a0014796

Carson, J. B., Tesluk, P. E., & Marrone, J. a. (2007). Shared Leadership in Teams: an Investigation of Antecedent Conditions and Performance. *Academy of Management Journal*, *50*(5), 1217–1234. doi:10.2307/20159921

Decuyper, S., Dochy, F., & Van den Bossche, P. (2010). Grasping the dynamic complexity of team learning: An integrative model for effective team learning in organisations. *Educational Research Review*, *5*(2), 111–133. doi:10.1016/j.edurev.2010.02.002

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2010). *THE SCRUM PRIMER*.

Dickinson, T. L., & McIntyre, R. M. (1997). A conceptual framework for teamwork measurement. In *Team Performance Assessment and Measurement: Theory, Methods, and Applications* (pp. 19–43). Retrieved from http://books.google.com.au/books?hl=en&lr=&id=lOo-Hq66FxkC&oi=fnd&pg=PA19&dq=a+conceptual+framework+for+teamwork+measurement&ots=HwQh1HqMDy&sig=PWfKVBVjW1trWTYD7RMlCroFA3c#v=onepage&q=a conceptual framework for teamwork measurement&f=false

Guzzo, R. a, & Dickson, M. W. (1996). Teams in organizations: recent research on performance and effectiveness. *Annual Review of Psychology*, *47*, 307–38. doi:10.1146/annurev.psych.47.1.307

Hadwin, A., & Järvelä, S. (2011). Introduction to a Special Issue on Social Aspects of Self-Regulated Learning : Where Social and Self Meet in the Strategic Regulation of Learning. *Teachers College Record*, *113*(2), 235–239.

Hoegl, M., & Gemuenden, H. G. (2001). Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence. *Organization Science*, *12*(4), 435–449. doi:1047-7039/01 /1204/0435/505.00 1526-5455

Kauffeld, S. (2006). Self-directed work groups and team competence. *Journal of Occupational and Organizational Psychology*, *79*(1), 1–21. doi:10.1348/096317905X53237

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, *33*(1), 159. doi:10.2307/2529310

Lane, K. L., Harris, K. R., Graham, S., Weisenbach, J. L., Brindle, M., & Morphy, P. (2008). The Effects of Self-Regulated Strategy Development on the Writing Performance of Second-Grade Students With Behavioral and Writing Difficulties. *The Journal of Special Education*, *41*(4), 234–253. doi:10.1177/0022466907310370

Langfred, C. W. (2004). Too Much of a Good Thing? Negative Effects of High Trust and Individual Autonomy in Self-Managing Teams. *Academy of Management Journal*, *47*(3), 385–399. doi:10.2307/20159588

Marks, M., & Panzer, F. (2004). The influence of team monitoring on team processes and performance. *Human Performance*, *17*(1), 25–41. Retrieved from http://www.tandfonline.com/doi/abs/10.1207/S15327043HUP1701_2

Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology*, *52*(5), 480–491. doi:10.1016/j.infsof.2009.11.004

Molenaar, I., & Chiu, M. M. (2013). Dissecting sequences of regulation and cognition: statistical discourse analysis of primary school children's collaborative learning. *Metacognition and Learning*, *9*(2), 137–160. doi:10.1007/s11409-013-9105-8

Molenaar, I., Sleegers, P., & van Boxtel, C. (2014). Metacognitive scaffolding during collaborative learning: a promising combination. *Metacognition and Learning*, *9*(3), 309–332. doi:10.1007/s11409-014-9118-y

Orton, J. D., & Weick, K. E. (1990). Loosely Coupled Systems : A Reconceptualization. *The Academy of Management Review*, *15*(2), 203–223.

Pfahl, D. (2014). Agline/Lean Methodes (pp. 1–12). University of Tartu. Retrieved from https://courses.cs.ut.ee/MTAT.03.094/2015_fall/uploads/Main/SE2014-handout11.pdf

Rising, L., & Janoff, N. S. (2000). The Scrum Software Development Process for Small Teams. *Ieee Software*, (August), 2–8.

Rogat, T. K., & Linnenbrink-Garcia, L. (2011). Socially Shared Regulation in Collaborative Groups: An Analysis of the Interplay Between Quality of Social Regulation and Group Processes. *Cognition and Instruction*, *29*(4), 375–415. doi:10.1080/07370008.2011.607930

Rogelberg, S. G., Barnes-Farrell, J. L., & Lowe, C. a. (1992). The stepladder technique: An alternative group structure facilitating effective group decision making. *Journal of Applied Psychology*, *77*(5), 730–737. doi:10.1037/0021-9010.77.5.730

Rousseau, V., & Aube, C. (2010). Team Self-Managing Behaviors and Team Effectiveness: The Moderating Effect of Task Routineness. *Group & Organization Management*, *35*(6), 751–781. doi:10.1177/1059601110390835

Santrock, J. W. (2011). *Educational Psychology* (5th ed., p. 584). New York: McGraw-Hill.

Schoor, C., Narciss, S., & Körndle, H. (2015). Regulation During Cooperative and Collaborative Learning: A Theory-Based Review of Terms and Concepts. *Educational Psychologist*, *50*(2), 97–119. doi:10.1080/00461520.2015.1038540

Van den Bossche, P., Gijselaers, W. H., Segers, M., & Kirschner, P. A. (2006). Social and Cognitive Factors Driving Teamwork in Collaborative Learning Environments. *Small Group Research*, *37*(5), 490–521. Retrieved from http://online.sagepub.com

Webb, N. M. (2009). The teacher's role in promoting collaborative dialogue in the classroom. *The British Journal of Educational Psychology*, *79*(Pt 1), 1–28. doi:10.1348/000709908X380772

Weinberger, A., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*, *46*(1), 71–95. doi:10.1016/j.compedu.2005.04.003

Whitworth, E., & Biddle, R. (2007). Motivation and Cohesion in Agile Teams. In *Agile Processes in Software Engineering and Extreme Programming* (pp. 62–69). Como, Italy: Springer Berlin Heidelberg. doi:10.1007/978-3-540-73101-6_9

Winne, P. H., & Nesbit, J. C. (2010). The psychology of academic achievement. *Annual Review of Psychology*, *61*, 653–78. doi:10.1146/annurev.psych.093008.100348