Motivation and Autonomy in Global Software Development

John Noll[1], Sarah Beecham[2], Abdur Razzak[2], Bob Richardson[3], Ann Barcomb[2], and Ita Richardson[2](✉)

[1] University of East London, Docklands Campus University Way, London, E16 2RD, UK
john.noll@uel.ac.uk
[2] Lero – the Irish Software Research Centre, University of Limerick, Limerick, Ireland
{sarah.beecham,abdur.razzak,ann.barcomb,ita.richardson}@lero.ie
[3] Ashling Microsystems Ltd., Limerick, Ireland
bob.richardson@ashling.com

Abstract. Global software development has become the norm rather than the exception for even the smallest companies. However, global software development is known to lead to numerous negative effects among distributed teams. This paper focuses on the effect of global software development on motivation. Specifically we ask: "Does increased autonomy, through the introduction of scrum, result in higher motivation among distributed developers?" We studied two distributed software development teams within one company headquartered in Ireland. Teams employ the scrum approach to software development which emphasizes, among other things, autonomy. We observed the teams during their scrum ceremonies, interviewed each team member and administered a motivation survey. We found that the difference in motivation levels before and after the introduction of scrum was slight and not statistically significant. Instead, there was a significant difference in the motivation levels of experienced team members, which were lower than less experienced members.

Keywords: Global software development · Agile software development · Scrum · Autonomy · Motivation

1 Introduction

As companies expand into new markets, acquire other companies in distant locations, and seek skilled staff in different locations, Global software development (GSD) has now become the norm rather than the exception. Even very small companies have developers and teams in remote locations.

Geographic separation, lack of timezone overlap, and cultural differences – collectively referred to as global distance – make the already complex task of software development even more complex. Geographic separation hinders informal communication that co-located teams use to clarify ambiguities and gaps in specifications and other formal documents. Lack of timezone overlap introduces communication delays that can slow progress, and in the extreme case prevents any kind of synchronous communication during normal working hours [1]. Cultural differences can introduce misunderstandings

I. Oshri et al. (Eds.): Global Sourcing 2017, LNBIP 306, pp. 19–38, 2017.
https://doi.org/10.1007/978-3-319-70305-3_2

as a result of different interpretations of requirements and other documents, and may cause mistrust due to misperceptions of different cultural norms. Organizations need to be aware of the negative impact that fear of losing control and jobs can have on the motivation of teams, thus decreasing the sharing of knowledge and levels of trust that can exist between colleagues [2].

Different organizational models have emerged to implement GSD, such as outsourcing, in-sourcing, near-shoring, off-shoring, etc. [3, 4]. Each of these approaches has specific needs for autonomy of the "remote" teams. Organizational boundaries, as exist for example in outsourcing arrangements, may require explicit contractual arrangements defining who can make what decisions about the project. However, other models also require good governance, because global distance renders conventional project management less effective [5, 6]. For example, a remote team in a distant timezone may experience substantial delays if they have to wait for the home office to make decisions. Through previous research in GSD, we observed that the different levels of autonomy were, somehow, presenting a difficulty for team members. This was supported by previous researchers in other disciplines. For example, in the Management literature, [7–9] have identified that a mismatch between an individual's need for autonomy, and the degree of autonomy someone actually has, can cause problems and may have an impact on motivation levels. We note from the Organisational Behaviour literature that motivation is viewed as a social process that defines how people join, remain part of, and perform adequately in, a human organization [10]. Motivation tends to be overlooked in project management since it is difficult to measure and control [11] due to its complex nature, yet motivation is shown to have an impact on the quality of work produced [12], productivity [13] and on employee retention [14]. Given that autonomy is strongly associated with job satisfaction [15], we postulate that members of teams who have less autonomy than they perceive to be necessary are less motivated. Furthermore, we are interested in the effect the introduction of scrum has on motivation. Given that scrum is expected to also increase autonomy within teams [16], our research question is: "Does increased autonomy, through the introduction of scrum, result in higher motivation among distributed developers?" We studied the motivation of members from two distributed project teams in a single company. This was undertaken within a larger software process improvement study in a medium-sized company in Ireland, where team members had made a transition from plan-driven to agile development (specifically scrum) just prior to our investigation.

This paper expands on research previously presented by [17] presented at the Global Sourcing Workshop, and by [18] at the Evaluation and Assessment in Software Engineering Conference, both in 2017. The paper is organized as follows: in the next section we give a brief background to Motivation theory in a global context, and reflect on changing software engineer characteristics. We discuss how GSD and Agile software development affects the software engineer, which motivates our research question. In Sect. 3 we present the case study, including our data collection and analysis methods. In Sect. 4 we present our qualitative and quantitative results. Section 5 discusses how our results address our research question. We conclude the paper in Sect. 6, with a summary of our findings, our limitations and plans for future work.

2 Background

There are numerous theories that try to explain the conscious or unconscious decisions people make to expend effort or energy on a particular activity [19]. These theories provide insight into what motivates software engineers to engage fully in their tasks, commit to the organization's goals, produce higher quality software [11], and stimulate innovation [20]. Conversely, a demotivated workforce can lead to project failure [21].

Table 1. Software engineer motivation factors [24]

Motivator	Type
Rewards and incentives	Extrinsic
Development/training needs addressed	Intrinsic
Variety of work	Intrinsic
Career Path	Intrinsic
Empowerment/responsibility/shared leadership	Intrinsic
Good Management	Extrinsic
Sense of belonging/team spirit	Extrinsic
Work/life balance	Extrinsic
Working in successful company	Extrinsic
Employee participation	Intrinsic
Feedback	Extrinsic
Recognition	Intrinsic
Equity	Intrinsic
Trust/respect	Intrinsic
Technically challenging work	Intrinsic
Job security/stable environment	Extrinsic
Identify with the task	Intrinsic
Autonomy	Intrinsic
Appropriate working conditions/infrastructure	Extrinsic
Making a contribution/task significance	Intrinsic
Sufficient resources	Extrinsic
Team quality	Extrinsic
Creativity/Innovation	Intrinsic
Fun (playing)	Intrinsic
Professionalism/setting standards	Extrinsic
Having an ideology	Extrinsic
Non-financial benefits (availability of rewards)	Extrinsic
Penalty Policies	Extrinsic
Good relationship with users/customers	Intrinsic
Recognition of cultural differences	Intrinsic
Recognition of individuality	Intrinsic
Construction/delivery/completion	Intrinsic

Of particular relevance to this study is Self Determination Theory, in which [15] postulate that to be self-motivated, employees require three innate psychological needs to be satisfied: the need for competence, autonomy, and relatedness.

2.1 Motivation in Software Engineering

Three reviews covering over 150 empirical studies of software engineer motivation [11, 22, 24] together with one case study [24] yield an inventory of 32 motivation factors for software engineers (Table 1). Among these are Problem Solving, Team Working, Change, Challenge, and Benefit. In addition, nine separate studies in the SLR undertaken by [11] on Software Engineer motivation identified autonomy to be an important factor.

2.2 Motivation and Agile

Traditionally, GSD has followed a plan-driven, structured, waterfall approach, where tasks are allocated according to where they appear in the software lifecycle [25]. It was considered that agile methods, envisaged for small projects and co-located teams with informal processes [26, 27], would be a poor fit for distributed development approaches [28] which relies on formal mechanisms. Yet, there is a growing trend for companies engaged in GSD to adopt agile methods [29, 30]. Adopting agile practices such as short iterations, frequent builds, and continuous delivery all pose challenges to configuration management and version management [31]. But, practices such as short iterations increase transparency of work-in-progress, and provide a big picture of project progress to stakeholders [32]. However, setting up an agile team is usually motivated by benefits such as increased productivity, innovation, and employee satisfaction [33].

Introducing agile methods can change the culture in a company - developers need to have more autonomy as well as decision-making power to implement agile practices [16]. Sutherland [34] states that autonomy is a key indicator that scrum is working, where "the scrum team is (and feels) totally responsible for their product and no outside agency impacts the work inside a sprint", while [35] have also found evidence of autonomy in the scrum teams whom they studied. Through frequent communications and meetings (i.e.; daily stand-ups), agile team members can motivate and influence each other's behavior [36], but little is known about motivation in an agile context [37, 38].

2.3 Motivation and Global Software Development

Some of the issues introduced by GSD [39] may be addressed by meeting the motivational needs of software engineers. For example, GSD projects have been shown to suffer from high staff turnover [40, 41] whereas high levels of motivation can have a positive effect on staff retention [14]. The review conducted by [22] and the case study by [23] looked at motivation of GSD software engineers. Both found that the GSD practitioner has specific and new needs, such as recognition of cultural differences and individuality, and the need to see how their work contributes to a complete and finished product.

2.4 Demotivation and Herzberg's Two Factor Theory

According to Herzberg's two factor theory [42], extrinsic motivators (also called hygiene factors) have the power to demotivate if absent, but when present do not trigger the long term desired impetus and positive energy of intrinsic motivators. Software engineers working in a multi-site team are likely to face many demotivating factors, which in turn can cause difficulties [43]. Among these are:

- Motivation and Autonomy in GSD;
- Inequity, where remote working causes training, growth and promotion opportunities to be missed, inequitable holiday allowances are given, and they may need to work anti-social hours to communicate with colleagues;
- Interesting work going to other parties, as complex tasks are retained at the home site, while less rewarding tasks such as maintenance go to the remote teams;
- Unfair reward system, which may happen if the remote software engineer is only noticed when there is a problem;
- Poor communication such as poor feedback, and loss of direct contact with other team members and management;
- Bad relationship with users and colleagues, where lack of face to face contact can result in mistrust and difficulty in building relationships with colleagues;
- Poor working environment, when being physically separated from the rest of the team, or the home site, is considered demotivating;
- Role ambiguity, which can occur when working in remote teams where each member is expected to take on many different roles, providing variety and challenge, but often resulting in overstretching the individual;
- Lack of influence, for example when senior management from the head office discusses issues with the client without involving the on-site project manager.

De-motivators, as listed above, are not necessarily the opposite of motivators, and so should be treated separately. For example role ambiguity is found to be de-motivating, but someone with a fixed job description may not be motivated. Sometimes, one factor can be both motivating and demotivating depending on context, e.g. working on maintenance tasks [11].

2.5 GSD Environmental Impact on Software Engineer Characteristics

A review of the literature found that in nearly three-quarters (73%) of the cases software engineers form a distinct identifiable occupational group [11]. Most cited characteristics were "growth oriented," "introverted," and "need for independence" indicating that these occur across many contexts. The view that software engineers are introverted reflects findings from [44] in their Job Diagnostics Survey. This view is not universal as some studies characterize software engineers as sociable people [11].

Although some research suggests that the needs of a global software engineer are similar to those of the general population of engineers [24, 45], speculated in their empirical study on software engineer motivation that this may be changing – in that working in distributed teams, the need to travel and less need for a work/life balance is attracting a different type of personality. In the same study, those engineers working in a GSD environment did not mention the following factors as attracting them to, or keeping in the software engineering field: development practices, autonomy, empowerment and responsibility, trust and respect, recognition of individuality. Of note here is that *autonomy* is not mentioned as important to the small sample of engineers in [24]. This distinguishing feature may be due to personality and "individual differences in their tendencies toward autonomous functioning across specific domains and behaviors" [46].

2.6 Is Autonomy Still an Important Factor?

We define autonomy as a feeling of independence, freedom and control (or self-determination) [7]. Autonomy has been identified in earlier studies as an important motivator for software engineers [11], and is also a core concept in self-determination theory [15, 47, 48]. [46] reason that the more autonomy one feels, the more intrinsically motivated one becomes. It might be that the global software engineer profile is changing as discussed in Sect. 2.4. This may reflect Deci and Ryan's Cognitive Evaluation Theory (CET) [7] that specifically addresses social and environmental external factors which facilitate or undermine intrinsic motivation. Taking this argument forward, and given that many environmental factors are inevitable when working in GSD (such as having to meet colleagues virtually, fitting in with hours of remote teams in different timezones, and travel), [24] suggest that those engineers who remain working in GSD teams for the long term are resilient to the demotivating factors that are inherent in GSD.

In this study, we focus on one factor, as picked up in [24] and ask whether autonomy affects motivation of software engineers working in GSD. Because autonomy is a crucial component of agile development, as well as important for software engineers' motivation, but potentially difficult to satisfy within the context of GSD, we examined the extent to which autonomy affects the motivation of two GSD teams within a company who were introducing scrum. Our research question is expressed as: "Does increased autonomy, through the introduction of scrum, result in higher motivation among distributed developers?" The research method is discussed in the next section.

3 Research Methods

As part of a larger software process improvement study, we studied two distributed software development teams: the first comprising six members in 3 locations in Wales, England and Ireland and the second with nine members in three locations in Ireland, Canada and USA. Both projects included teams comprising former employees of companies acquired by the current parent company. We observed team planning, review meetings, and daily "scrums," over a period of ten months for one team and fifteen months for the second team. We also interviewed each team member, and asked them to rate their motivation on a five likert point scale. Finally, we asked all members of each team to complete a short motivation survey.

3.1 The Case

The company we studied, which we shall call OptiManage, is a medium-sized Irish-based software company that develops practice management software for the optical industry. The business model which the company has been using is to acquire small companies worldwide. When companies are acquired, they become part of the global software team, thus integrating their software processes with that of OptiManage. The software developed by the acquired company is supported by OptiManage until it is either absorbed into the main product or phased out completely. This has resulted in OptiManage having a headquarters in Dublin, Ireland employing approximately 50 software engineers, and many small GSD teams. Teams hold face-to-face meetings at least twice per year, conducting their global, often daily, interactions through means such as video conferencing, e-mail and messaging.

We studied two teams, each of which is involved in different aspects of OptiManage's business. Team A is responsible for maintaining the core software for their product line. They also maintain and enhance the retail product for the Irish, UK, Canadian, and Mexican markets. Finally, they perform maintenance on a legacy product resulting from an acquisition that also brought four of Team A's team members to the company. Team A's members are located in England, Wales, and Ireland. Two of Team A's members work primarily from home in England, the other members are distributed equally between the head office in Ireland and an office in Wales. For this team, they are working in similar time zones and, as everyone speaks English, they do not have to deal with language barriers. Team B is distributed between Ireland, and the west coasts of Canada and USA. Within each of these countries the members are co-located. Their responsibility is to tailor the company's product for a large customer in North America. While there is no language barrier – all team members speak English as their native language, this team have to cope with up to eight hours difference in timezones. Team Composition (Table 2) shows the distribution of members of both teams. Both teams use Agile Software Development methods, holding scrum "ceremonies" including daily stand-up, sprint planning and retrospective meetings. The Project Manager also plays role of Scrum Master.

3.2 Data Collection

Two of the authors acted in a participant-observer role by sitting in on each team's scrum "ceremonies". Team A was observed from November, 2015 to June, 2016, and Team B was observed from January, 2016 to March, 2017. Due to the team being global, they held video conferences for daily standups, sprint planning, backlog grooming, and sprint retrospectives. The observer also conducted semi-structured interviews with each member of the team (see Appendix A). The interview protocol was based on that used by [49], and was extended in this project to include questions triggered during the participant-observation sessions within OptiManage.

Table 2. Case study team composition

Country		Number
Team A		**6**
Ireland	Software developer	2
Wales	Scrum master	1
	Product owner	1
England	Quality assurance	1
	Senior Developer	1
Team B		**9**
Ireland	Product owner	1
	Software developer	3
	Quality assurance	1
Canada	Scrum master	1
	Product owner	1
	Developer	1
USA	Senior developer	1

All respondents were asked to describe their backgrounds, roles on the team, and development processes. They were asked to rate his or her motivation on a five point interval scale - definitely low (1), somewhat low (2), neither low nor high (3), somewhat high (4) and definitely high (5). In addition, as participant-observers, each researcher kept a journal of the daily ceremonies, which was retained in note form for future reference. The interviews were recorded and later transcribed.

Finally, we held a workshop in early 2016, in which all Team A members attended in person. It was based in Dublin, at the head office. During this workshop Team A completed a short motivation survey (see Appendix A – A.4). Team B completed the survey via e-mail. We have administered this survey with other software development companies operating across geographic boundaries, see, for example, the case study described in [24]. The survey is an adaptation of questions created by [50] designed specifically to reveal what motivates practicing software engineers.

3.3 Data Analysis

We used a mixed methods analysis approach, where quantitative methods were used to rate and aggregate the levels of motivation, and qualitative methods were applied to explore themes in the semi-structured interview data, observations, and in the responses to the open questions in the motivation survey.

Quantitatively, results were aggregated across individuals and teams based on a 5 point Likert scale to gain a measure of motivation. We also aggregated the responses to the motivation questions in the survey. Qualitatively, we took an inductive approach, and analyzed the responses to the open questions in Sections A3–A5 of the survey, grouping the survey responses into themes using content analysis [51]. We also took a deductive approach, by looking specifically for themes coming from the semi-structured interviews and observations that related to autonomy.

4 Results

We first present results of team members' self-reported motivation. As mentioned in the previous section, at the end of each interview, interviewees were asked to rate their motivation on a five-point interval scale (see Table 3). There is little difference of motivation between roles. Two developers reported "definitely high" motivation, two reported "somewhat high" motivation, and four reported "neither low nor high" motivation. One product owner reported "definitely high" motivation, the other two "neither low nor high" motivation. The two scrum masters and the two quality assurance team members reported either "neither low nor high" or "somewhat high" motivation. Developers seem to be more motivated after the introduction of scrum. However, once again the Wilcoxon Signed-Rank test does not show this difference to be significant (p-value = .24). Location seems to favor the home office or the most remote locations in North America (Canada and USA): each had a median motivation rating of "somewhat high", while North America also had a mode of "somewhat high". Similarly, the home office and North America both appear to have slightly increased motivation after the introduction of scrum, but the Wilcoxon Signed-Rank test does not show this difference to be significant (p-value = .5 and .68 respectively).

Table 3. Team member motivation by role

Country	Number	Minimum	Maximum	Median	Mode
Scrum master /Project manager	2	3	4	3.5	3&4
Product owner	3	3	5	3	3
Developer	8	3	5	3.5	3
Quality assurance	2	3	4	3.5	3&4

Table 4 summarizes the results from before and after the introduction of scrum. (Note: three additional team members were hired after scrum was introduced). The range of reported motivation ranges from "neither low nor high" to "very high" motivation, both before and after scrum introduction - no-one reported low motivation. The most common motivation level before the introduction of scrum was "neither low nor high", while after the introduction of scrum, motivation levels were evenly distributed among "neither low nor high," "somewhat high," and "definitely high," with some team members reporting higher motivation, others reporting lower, and some reporting no difference. This suggests that introducing scrum had a positive effect on motivation (see Fig. 1). However, comparing the before and after results using the Wilcoxon Signed-Rank test for differences between populations shows no statistically significant difference (p-value = .4) between the motivation levels of team members that were present before and after the introduction of scrum.

Table 4. Team member motivation by location, before and after introduction of scrum

Location/Scrum stage	Rating	Total	Median	Mode
	3 4 5			
Ireland/before	2 1 2	5	4	3&5
Ireland/after	1 3 3	7	4	4&5
UK/before	2 2 0	4	3.5	3&4
UK/after	3 0 1	4	3	3
Nth America/before	1 1 1	3	4	3,4,5
Nth America/after	1 2 1	5	4	4

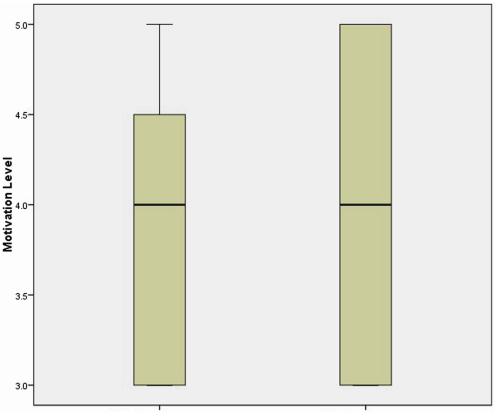

Fig. 1. Individual motivation before (left) and after (right) the introduction of scrum

Finally, experience does seem to affect motivation: the highest motivation scores were reported by the team members with less than ten years' experience (Tables 5 and 6, Fig. 2). Comparing these less experienced developers to their peers with ten or more years' experience, the Wilcoxon Signed-Rank test does not show any difference in motivation before the introduction of scrum. However, after the introduction of scrum, the less experienced developers did have significantly higher motivation (p-value = .04 for the unpaired Wilcoxon test[1] (Fig. 2).

Table 5. Team member motivation by experience

Years of Experience	Number	Minimum	Maximum	Median	Mode
<5 years	2	3	4	3.5	3&4
5–9 years	5	3	5	3.5	3
10–19 years	5	3	5	3	3
20+ years	3	3	5	3.5	3

[1] We used the unpaired Wilcoxon test to compare two different samples, rather than pairs of results from the same sample.

Table 6. Team member motivation by experience, before and after introduction of scrum

Experience/Scrum stage	Rating	Total	Median	Mode
	3 4 5			
<10 years/before	2 1 1	4	3.5	3
<10 years/after	0 4 3	7	4	4
10+ years/before	3 3 2	8	4	3&4
10+ years/after	5 1 2	8	3	4

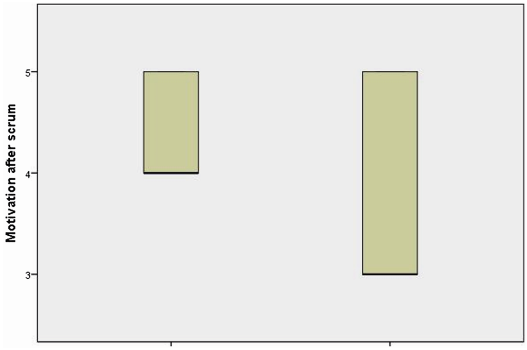

Fig. 2. Individual motivation after scrum implementation showing less than 10 years' experience (left) and greater than ten years' experience (right).

5 Discussion

Examining our qualitative responses, looking at motivation levels, some participants divided their level of motivation according to current role and current project. Three gave a measure of their personal level of motivation: two stating it was 'very high' (Respondents 1 & 4), and one 'somewhat high' (Respondent 6). In contrast when rating their motivation on current project, three responded 'somewhat high' (Respondents 1, 5 and 6). This shows a slight shift from 'very high' personal motivation, to lower motivation in this particular project. The two that gave a general overall level of motivation, stated their motivation level was 'neither high nor low' (Respondents 2, and 3).

One of the attributes presented as a positive for agile software development, within which scrum is a technique, is that teams have more autonomy than in plan-driven e.g. waterfall development [16]. In our study, we observed that less experienced developers ('Developers' in tables in Sect. 4) have increased levels of motivation – each one returning a 'somewhat high' (4) or 'definitely high' (5) level. The numbers are small, and there is no significant difference in motivation levels before and after the introduction of scrum. However, if we observe the more experienced engineers, we note that the majority returned "neither low nor high" (3) levels of motivation after scrum introduction. We also note that the Ireland/England/Wales teams registered lower levels of motivation than the Ireland/Canada/USA team.

One possible reason for the lower motivation among the more experienced developers might be due to these developers having less autonomy than would normally be found in an agile environment. Our findings showed hints of issues concerning autonomy in responses from two more experienced respondents. They described issues like "*being overruled by seniors*" and "*team decisions disregarded by higher management.*" This may be GSD, location, company or scrum-specific. In their paper, in which they studied a different global company, [24], noted that 'autonomy' was not a motivator expressed in survey responses. This may also be the case here, and warrants further investigation to understand whether GSD software engineers demonstrate different motivators than those working in co-located situations.

In contrast, the more junior members were comfortable with their dependence on senior developer inputs in the planning. One of the Canadian developers on Team B reported that all of his code had to be reviewed by a senior developer in Dublin, suggesting a lack of autonomy. Yet the same developer said that "*checks and balances*" and "*more communication*" which are hallmarks of scrum, result in a "*better product at the end of the sprint*", suggesting that he did not perceive a mismatch between his ideal and actual autonomy.

This reflects a healthy attitude as noted in earlier work with a high performing agile team where decisions were made by consensus, and when asked about what drives down performance, the high performing team members responded "developers wanting to do things the way they want to and not listen to anyone else" [53]. Developers in our sample were exhibiting similar behaviour to those developers in the high performance team.

Another factor regarding motivation within scrum teams is the customer. One developer, based in Wales, stated that she feels "*very motivated now*", despite difficulties due to customer dynamics as they "*sometimes neglects the important tasks at hand*". Another developer who rated their motivation as "somewhat high" was concerned that he had to "*humour every single request no matter how obscene*". The participants who rated their motivation as "neither low nor high" liked the idea that they do not have to deal with customers, and so possibly enjoyed a level of autonomy where they were allowed to focus their own programming activities.

Other reasons for high motivation were cited, for example, a senior developer in Dublin mentioned intellectual challenge "*To be honest as long there is new stuff to do or new task to do it [the process] doesn't matter. So, it's very high.*"

Finally, a comment from one experienced Team A member sheds light on what might be the true reason for some of his lack of motivation. Describing Team A's role as maintainers of the core codebase, this person said: "*It has to be done, but nobody else wants to do it.*" Maintenance tasks have been found to be de-motivating [11], and may even be overlooked in process improvement activities [52].

These results support our hypothesis that software engineers in GSD teams do not perceive a lack of autonomy nor reduced motivation stemming from it even when they appear to have less autonomy than is generally expected in an agile environment. It also supports the idea that engineers who persist with GSD are less negatively affected by aspects of GSD than other engineers [24].

Clearly these observations are based on limited evidence, and there may be several other factors that are influencing the levels of motivation that we are not currently measuring, or may not be able to measure. However, our study starts to build a hypothesis where the global software engineer has an awareness of the dependence they have on their team members and management, and are not so concerned with a lack of autonomy.

It could be that autonomy alone is not sufficient, and that it needs to be matched by creativity, identity and variety. Perhaps the more experienced software engineers in our sample were not able to be creative, and their work lacked variety. Identity in terms of recognition for a task well done can lack visibility outside the team, a problem we identified in an earlier study when analysing motivation in a high performing team [37]. This also supports the findings in [24] in which creativity, construction, and making a contribution was by far the most prevalent motivation factor mentioned across the sample of experienced engineers. Do these results support our hypothesis that software engineers characteristics are changing? If our hypothesis is correct, then, in addressing our research question, a perceived lack of autonomy may be less important to the global software engineers in our study.

It could also be that the level of autonomy which exists in co-located scrum is not evident in GSD teams. [24] also identified in their previous study that 11 software engineering motivation factors did not exist in the GSD team they studied. One of these was autonomy. The study presented in this paper is also pointing in this direction. However, amongst the many answers to the open motivation and de-motivation questions, very few related to autonomy. The participants who rated their motivation as neither high nor low liked the idea that they do not have to deal with customers and possibly enjoyed a level of autonomy, where they were allowed to get on with their own programming activities. But this is just conjecture at this stage. What it does say is that we need to re-structure our motivation questionnaire to focus on those factors which may not be present as motivators for GSD software engineers.

Our motivation survey results for Team A reveal further potential reasons for their lower levels of motivation. Firstly, focusing on the second section of the survey "What motivates you?", some respondents slightly misunderstood question 6, in that they divided their motivation levels according to their current role (personal level), and the motivation in working on the project. This error was actually revealing, and suggests that we need to re-design the survey in future. The motivation in working on the project was slightly lower than their personal levels of motivation (this is consistent with the semi-structured interview findings), supporting the idea that they don't get intrinsic motivation from undertaking maintenance tasks (which is core to their work) or that there are certain pressures on the current project that is reducing their motivation. Looking at the responses to the open questions on what motivates this group, their answers support previous research, in that intrinsic motivations are what attract them to software engineering as a profession. Factors include making a difference, problem solving, constructing something from nothing, and learning something new.

The participant, who gave a very high level of motivation rating, noted that the Team and Support were important to them. Respondent (5), was very aware of the dependence on senior developer inputs into the planning – and found this de-motivating since a lack of input in planning breaks the sprint. This respondent was suggesting that they needed more support from the senior developer, rather than working alone. This is re-iterated when they suggest that involving other senior developers into the sprint grooming planning was an issue. Respondent 5 had a ‘somewhat high’ level of motivation.

We are aware that in previous studies, e.g. [54], there has been some discussion around scrum and agile methods offering “no advice on how shared leadership could be implemented”. There is a possibility that these teams need guidance on how this should happen within the context of agile implementation. Indeed, in their research, [55] concluded that there is a need for team leaders to have development programs “aimed at developing capabilities for adaptive switching of achievement priorities, and for effectively communicating changing goal priorities”. The introduction of scrum in the versatile business and software development environment in which OptiManage operates.

5.1 Threats to Validity

We note that our study has limitations which threaten the validity of our conclusions. Our 5-point measure for motivation was only used once during the study period. However, we asked 15 team members at different levels, and have supplemented their answers with qualitative research which has given insight into why those studied presented the specific answers.

Also, we did not have a direct measure of autonomy. Because scrum implementation is expected to introduce autonomy into teams [16], we assumed this would work also for global software development teams. This, of course, raised the additional question for us to study – in global software development teams, does scrum provide autonomy to team members? We also note that the study participants were from one company, and therefore, this factor may be company related. However, our research to date provides no evidence to support this. Given that our study consisted of only fifteen subjects, we have not generalized our findings.

6 Conclusion

Prior research has shown that software engineers who are motivated deliver higher quality software [11], are more innovative [19], more successful [20] and less prone to attrition [14]. Companies, including those with GSD teams, are adopting agile methods [56] in an effort to realize benefits such as increased productivity, innovation, and employee satisfaction [32]. However, agile methods were originally designed for small, co-located teams [25, 26], and require significant autonomy to be fully deployed [16].

Following this study, we have identified a number of questions which we need to consider:

Why are there apparent differences in motivation between the two teams? To uncover this, we intend to administer the motivation survey developed by [50] to all project groups (currently only Teams A and B have completed the survey). We would like to identify whether this effect is company specific, or whether there is a possibility that the autonomy expected from scrum implementation is lessened when it is implemented in global software development teams.

Could gender difference be a factor in motivation? We have not split the results by gender, but 33% of Team A was female developers and 22% of Team B were female. While this is a higher percentage than often seen in software development teams globally (~10%), given that it is now realized that women have different requirements than men in work situations, it would be interesting to investigate this further.

Are the software engineer motivation factors identified in [24] different for software engineers who are co-located and software engineers working GSD teams? This requires the distribution of motivation surveys developed by [50] to co-located and GSD teams in multiple countries. In doing this, there needs to be account taken of other potential factors, such as type of work being completed, culture and responsibility of the person completing the survey.

When implemented in GSD, does autonomy exist to a lesser extent than when implemented in co-located teams? Again, a detailed study is needed, supported by the development of a measure of autonomy to compare against motivation level.

Our study has found little evidence to suggest there is a difference in motivation between members of agile teams, and those in teams employing plan-driven development. Our research has shown that motivation differences related to experience, intellectual challenge and contributing to a valuable product.

Scrum emphasizes "self-organizing teams" that decide among themselves the best way to achieve their objectives. As such, we expected that motivation would be higher after the introduction of scrum due to higher autonomy. We found, however, that the difference in motivation levels before and after the introduction of scrum was slight and not statistically significant. Instead, there was a significant difference in the motivation levels of experienced team members, which were lower than less experienced members. We speculate that this is due to the absence of other motivators that are important to senior-level software engineers. We conclude that, while autonomy is an important motivator, it is not sufficient on its own, and that the implementation of scrum within GSD may not provide the autonomy level expected and seen in co-located teams.

Acknowledgments. We thank the members of Team A and Team B for their generous and thoughtful collaboration on this study, and OptiManage, for allowing us to study their software development efforts. This work was supported, in part, by Science Foundation Ireland grants 10/CE/I1855 and 13/RC/2094 to Lero - the Irish Software Research Centre (http://www.lero.i.e). This work was supported, in part, by Enterprise Ireland and the European Development Fund through grant IR20130022.

Appendix A: Interview Protocol

A.1 Demographics

(1) Time at OptiManage.
(2) Time on current project.
(3) Current position.
(4) Current location.
(5) Previous position & company.

(6) Total development experience.
(7) Total domain experience.
(8) Education and formal qualifications.
(9) Gender.
(10) Nationality.

A.2 Motivation Rating

For the next two questions, rate your motivation on the following scale: Very low, Somewhat low, Neither low nor high, Somewhat high, Very high

(1) How would you rate your motivation now?
(2) How would you rate your motivation prior to introduction of Scrum?

A.3 Project

(1) How would you describe your current project?
(2) How would you describe your project's current domain?
(3) What is your role?
(4) Have you met any of your remote colleagues?
(5) Does geographic separation hinder the project?
(6) Why do you think OptiManage is employing distributed devel-opment for this project?
(7) Have you had any training in Agile methods?
(8) Have you had any training in distributed development?

A.4 Process

(1) Describe your dev process.
  (a) How do devlopers test changes?
  (b) How are builds created for QA?
  (c) Is build machine a bottleneck?
  (d) How does "outside of sprint" work?
  (e) Are there separate QA tasks?
  (f) Are spikes time-boxed? How is effort accounted for?
  (g) Is sprint too short?
  (h) Does the Product Owner ever make estimates?
  (i) What is the [important customer] button?
  (j) Who is [Chief Architect]?
(2) Does the application domain restrict the process in any way?
(3) How would you describe the previous process (before Scrum)?
(4) What advantages does Scrum offer over that process?
(5) What advantages did the previous process have over Scrum?
(6) What is working well with Scrum?
(7) What is not working well with Scrum?
(8) What obstacles exist that prevent Scrum from working well?
(9) What is the best aspect of Scrum?
(10) What is the worst aspect of Scrum?

(11) If there was one thing you could change, what would it be?

A.5 Motivation

1. What aspects of your work in software engineering do you get most satisfaction from?
2. What makes you stay working in software engineering?
3. What factors attracted you to work in software engineering?
4. What makes software development worthwhile to you?
5. What do you plan to do when you have completed your current project?
6. On a scale of 1 – 5 how motivated are you in your current role and project? (1 = very low, 2 = somewhat high, 3 = neither high nor low, 4 = somewhat high, 5 = very high).
7. If you didn't answer 5 to the previous question, what do you think could improve your motivation?
8. Are there any aspects of your job that you find de-motivating? If so, please list the top three here.
9. Please add any ideas you have here about motivating or de-motivating aspects of your job. (Note: motivating aspects of your job relate to things that you do for reasons of personal fulfillment. De-motivating aspects are constraints that are external to you and outside your immediate control).

References

1. Casey, V., Richardson, I.: Practical Experience of Virtual Team Software Development, European Software Process Improvement Conference, Industrial Proceedings, Trondheim, Norway, pp. 3–D.9-13-D.15, 9–11 November 2004
2. Casey, V., Richardson, I.: Virtual teams: understanding the impact of fear, special issue on global software engineering, J. Softw. Evol. Process **13**, 511–526 (2008)
3. Smite, D., Wohlin, C., Galvin, Z., Prikladnicki, R.: An empirically based terminology and taxonomy for global software engineering. Empirical Softw. Eng. **19**(1), 105–153 (2014)
4. Vizcaıno, A., Garcıa, F., Piattini, M., Beecham, S.: A validated ontology for global software development. Comput. Stand. Interfaces **46**(2016), 66–78 (2016)
5. Noll, J., Beecham, S., Richardson, I., Nic Canna, C.: Global teaming for global software development governance: a case study. In: International Conference on Global Software Engineering. IEEE, Irvine (2016)
6. Deshpande, S., Richardson, I.: Management at the outsourcing destination – global software development in india. In: 4th IEEE International Conference on Global Software Engineering, Limerick, pp. 217–225, 16–19 July 2009
7. Deci, E.L., Ryan, R.M.: Cognitive Evaluation Theory Intrinsic Motivation and Self-Determination in Human Behavior, pp. 43–85. Springer, US (1985). doi:10.1007/978-1-4899-2271-7_4
8. Ferratt, T.W., Harvey, G.E., Prasad, J.: Instrument validation for investigating a model of employment arrangement fit for IT professionals. In: Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research: Freedom in Philadelphia–leveraging Differences and Diversity in the IT Workforce (SIGMIS CPR 2003). ACM, New York, pp. 168–178 (2003)

9. Porter, L.W., Pearce, J.L., Tripoli, A.M., Lewis, K.M.: Differential perceptions of employer's inducements: implications for psychological contracts. J. Organ. Behav. **19**(1998), 769–782 (1998)
10. Huczynski, A., Buchanan, D.: Organizational Behaviour: An Introductory Text. (1991)
11. Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H.: Motivation in software engineering: a systematic literature review. Inf. Softw. Technol. **50**(9), 78–860 (2008)
12. Boehm, B.W.: Software engineering economics, vol. 197. Prentice-hall Englewood Cliffs (NJ) (1981)
13. Law, A., Charron, R.: Effects of agile practices on social factors. In: ACM SIGSOFT Software Engineering Notes, vol. 30. ACM, pp. 1–5 (2005)
14. Hall, T., Beecham, S., Verner, J., Wilson, D.: The Impact of staff turnover on software projects: the importance of understanding what makes software practitioners tick (refilling the pipeline: meeting the renewed demand for information technology workers). In: ACM-SIGMIS CPR 2008 Conference. Charlottesville, Virginia, 3-5 April 2008
15. Ryan, R.M., Deci, E.L.: Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Am. Psychol. 55(1), 68–78 (2000) http://dx.doi.org/10.1037/0003-066X.55.1.68
16. Fowler M.: Using an agile software process with offshore development. Capturado em, May 2006 http://martinfowler.com/articles/agileOffshore.html
17. Noll, J., Richardson, B., Razzak, M.A., Barcomb, A., Richardson, I., Beecham, S.: "Motivation and Autonomy in Global Software Development." The 11th Global Sourcing Workshop in La Thuile, Italy (2017)
18. Noll, J., Razzak, M.A., Beecham, S.: Motivation and autonomy in global software development: an empirical study. In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering (EASE 2017). ACM, New York, pp. 394–399 (2017) https://doi.org/10.1145/3084226.3084277
19. Petri, H.L., Govern, J.M.: Motivation: Theory, Research, and Application -, 6th edn. Wadsworth Publishing, Belmont (2012)
20. Frey, B.S., Osterloh, M.: Successful Management by Motivation: Balancing Intrinsic and Extrinsic Incentives. Springer-Verlag, Berlin (2002). doi:10.1007/978-3-662-10132-2
21. Verner, J.M., Babar, M.A., Cerpa, N., Hall, T., Beecham, S.: Factors that motivate software engineering teams: a four country empirical study. J. Syst. Softw. **92**, 115–127 (2014)
22. França, A.C.C., Gouveia, T.B., Santos, P.C.F., Santana, C.A., da Silva, F.Q.B.: Motivation in software engineering: a systematic review update. In: 15th Annual Conference on Evaluation Assessment in Software Engineering (Ease 2011), pp. 63–154 (2011). doi:10.1049/ic.2011.0019
23. de, Farias, Jr., I.H., Duarte, L., de, Oliveira, J.P.N., Dantas, A.R.N., Barbosa, J.F., de, Moura, H.P.: Motivational factors for distributed software development teams. In: 2012 IEEE Seventh International Conference on Global Software Engineering Workshops (ICGSEW), pp. 49–54. IEEE (2012). doi:10.1109/ICGSEW.2012.17
24. Beecham, S., Noll, J.: What Motivates Software Engineers Working in Global Software Development?, pp. 193–209. Springer International Publishing, Berlin, Heidelberg (2015). doi:10.1007/978-3-319-26844-6_14
25. Estler, H.C., Nordio, M., Furia, C.A., Meyer, B., Schneider, J.: Agile vs. structured distributed software development: a case study. Empirical Softw. Eng. **19**(5), 224–1197 (2014)
26. Kahkonen, T.: Agile methods for large organizations-building communities of practice. In: Agile Development Conference, pp. 2–10. IEEE, 22 Jun 2004
27. Abrahamsson, P., Conboy, K., Wang, X.: Lots done, more to do': the current state of agile systems development research. Eur. J. Inf. Syst. **18**(4), 281 (2009)

28. Ramesh, B., Cao, L., Mohan, K., Xu, P.: Can distributed software development be agile? Commun. ACM **49**(10), 6–41 (2006)
29. Hanssen, G.K., Šmite, D., Moe, N.B.: Signs of agile trends in global software engineering research: a tertiary study. In: 2011 Sixth IEEE International Conference on Global Software Engineering Workshop (ICGSEW), pp. 17–23. IEEE, 15 Aug 2011
30. Fitzgerald, B., Stol, K.J., O'Sullivan, R., O'Brien, D.: Scaling agile methods to regulated environments: an industry case study. In: 2013 35th International Conference on Software Engineering (ICSE), pp. 863–872. IEEE, 18 May 2013
31. Paasivaara, M., Lassenius, C.: Could global software development benefit from agile methods?. In: International Conference on Global Software Engineering, ICGSE 2006, pp. 109–113. IEEE 2006
32. Paasivaara, M., Lassenius, C.: Using iterative and incremental processes in global software development. In: 3rd International Workshop on Global Software Development, pp. 42–47, 24 May 2004
33. Šmite, D., Moe, N.B., Ågerfalk, P.J.: Fundamentals of agile distributed software development. In: Agility Across Time And Space 2010, pp. 3–7. Springer, Berlin (2010)
34. Sutherland, J.: Future of scrum: parallel pipelining of sprints in complex projects. In: 2005 Proceedings of Agile Conference. IEEE (2005)
35. Monteiro, C.V., da Silva, F.Q., dos Santos, I.R., Farias, F., Cardozo, E.S., do A Leitão, A.R., Pernambuco Filho, M.J.: A qualitative study of the determinants of self-managing team effectiveness in a scrum team. In: Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering, pp. 16–23. ACM, May 2011
36. Das, T.K., Teng, B.S.: Trust, control, and risk in strategic alliances: an integrated framework. Organ. stud. **22**(2), 83–251 (2001)
37. Beecham, S., Sharp, H., Baddoo, N., Hall, T., Robinson, H.: Does the XP environment meet the motivational needs of the software developer? an empirical study. In: Agile Conference (AGILE), pp. 37–49. IEEE 13 Aug 2007
38. McHugh, O., Conboy, K., Lang, M.: Motivating agile teams: a case study of teams in Ireland and Sweden. In: International Research Workshop on IT Project Management 2010
39. Noll, J., Beecham, S., Richardson, I.: Global software development and collaboration: barriers and solutions. ACM inroads **1**(3), 66–78 (2010)
40. Holmstrom, H., Ó Conchúir, E., Ågerfalk, P.J., Fitzgerald, B.: Global software development challenges: a case study on temporal, geographical and socio-cultural distance. In: International Conference on Global Software Engineering ICGSE 2006, pp. 3–11. IEEE (2006)
41. Ebert, C., Murthy, B.K., Jha, N.N.: Managing risks in global software engineering: principles and practices. In: IEEE International Conference on Global Software Engineering (ICGSE 2008), pp. 40–131. IEEE (2008)
42. Herzberg, F., Mausner, B., Snyderman, B.B.: Motivation to Work, 2nd edn. Wiley, New York (1959)
43. Beecham, S.: Motivating software engineers working in virtual teams across the globe. In: Wohlin, C., Ruhe, G. (eds.) Software Project Management in a Changing World, pp. 82–255. Springer, Germany (2014). doi:10.1007/978-3-642-55035-5_10
44. Couger, J.D., Zawacki, R.A.: Motivating and Managing Computer Personnel. Wiley, New York (1980)
45. El Khatib, V., Trang, S., Reimers, K., Kolbe, L.: The role of motivational factors in distributed software development teams: an empirical investigation. In: ECIS, p. 221 (2013)

46. Ryan, R.M., Deci, E.L.: Self regulation and the problem of human autonomy: does psychology need choice, self determination, and will ? Journal of Personality, 74:6. Blackwell Publishing, Inc. DOI:10:1111/j.146-6494.2006.00420.x. (2006). http://www.intrinsicmotivation.net/SDT/documents/2006_RyanDeci_Self-RegulationProblemofHumanAutonomy.pdf
47. Deci, E.L., Ryan, R.M.: Intrinsic Motivation and Self-Determination in Human Behavior. Plenum Press, New York (1985). doi:10.1007/978-1-4899-2271-7
48. Gagne, M., Deci, E.L.: Self.-determination. theory work. motivation. J. Organ. Behav. **26**, 331–362 (2005). John Wiley and Sons Ltd
49. Beecham, S.: Motivating software engineers working in virtual teams across the globe. In: Ruhe, G., Wohlin, C. (eds.) Software Project Management in a Changing World, pp. 255–277. Springer, Heidelberg (2014). doi:10.1007/978-3-642-55035-5_10
50. Sharp, H.: “What Motivates Software Engineers: A Workshop Report.” Overload Project Management, p. 99 (2010). http://accu.org/index.php/journals/1703
51. Krippendorff, K.: Content Analysis: an Introduction to Its Methodology, 2nd edn. Sage Publications, Beverly Hills (2004)
52. Hall, T., Rainer, A., Baddoo, N., Beecham, S.: An empirical study of maintenance issues within process improvement programmes in software industry. In: IEEE Conference Soft Maintenance, Florence, Italy, pp. 422–423, 7-9 November 2001
53. Beecham, S., Sharp, H., Baddoo, N., Hall, T., Robinson, H.: Does the XP environment meet the motivational needs of the software developer an empirical study. In: Agile Conference (AGILE 2007), pp. 37–49. IEEE (2007)
54. Moe, N.B., Dingsøyr, T., Dybå, T.: A teamwork model for understanding an agile team: a case study of a scrum project. Inf. Softw. Technol. **52**(5), 480–491 (2010)
55. Alexander, L., Van Knippenberg, D.: Teams in pursuit of radical innovation: a goal orientation perspective. Acad. Manag. Rev. **39**(4), 423–438 (2014)
56. Hanssen, G.K., Smite, D., Moe, N.B.: Signs of agile trends in global software engineering research: a tertiary study. In: Sixth IEEE International Conference on Global Software Engineering Workshop (ICGSEW), pp. 17–23. IEEE (2011)

Springer

http://www.springer.com/978-3-319-70304-6

Global Sourcing of Digital Services: Micro and Macro
Perspectives
11th Global Sourcing Workshop 2017, La Thuile, Italy,
February 22-25, 2017, Revised Selected Papers
Oshri, I.; Kotlarsky, J.; Willcocks, L.P. (Eds.)
2017, VII, 163 p. 25 illus., Softcover
ISBN: 978-3-319-70304-6