

What makes Software Engineers go that extra mile?

Rien Sach
*Maths and Computing
Faculty
Open University
Milton Keynes, MK7 6AA
United Kingdom
r.j.sach@open.ac.uk*

Helen Sharp
*Maths and Computing
Faculty
Open University
Milton Keynes, MK7 6AA
United Kingdom
h.c.sharp@open.ac.uk*

Marian Petre
*Maths and Computing
Faculty
Open University
Milton Keynes, MK7 6AA
United Kingdom
m.petre@open.ac.uk*

Keywords: POP-II.A. Professional; POP-V.B. Interviews; POP-VI.F. Exploratory.

Abstract

Current research suggests that motivation factors in software engineering are evolving and that our current understanding of motivation in software engineering is out of date. In this paper an analysis of semi-structured interviews with 13 professional software engineers is presented. The data suggests that ‘the work’ makes software engineers go that extra mile and that ‘people’ are important, but that ‘obstacles’ really sap the energy of software engineers.

1. Introduction

Motivation has been reported as having an impact in software engineering on productivity (Procaccino *et al.* 2009), software quality (Boehm 1981), and the overall success of the project (Hall *et al.* 2008). Motivation has also been commonly cited as a cause of software development project failure (DeMarco and Lister 1999).

A systematic literature review of motivation in software engineering conducted by Beecham *et al.* (2008) analysed 92 papers published before March 2006 and identified 21 motivators and 15 de-motivators present in the literature. This research was extended by Franca *et al.* (2011) who analysed 53 papers covering March 2006 to August 2010 using an approach heavily based on the Beecham *et al.* (2008) research.

This recent work by Franca *et al.* (2011) identified 8 additional motivators (team quality, creativity/innovation, fun, professionalism, having an ideology, non-financial benefits, penalty policies, and good relationship with users/customer), while also stating that two of the original motivators from the Beecham *et al.* (2008) study were not present (appropriate working conditions, and sufficient resources). Franca *et al.* (2011) identified one additional de-motivator from the reviewed literature (task complexity).

The addition of 8 motivators and 2 de-motivators supports claims that the motivation of software engineers has evolved since the majority of research was conducted (Sharp and Hall 2009), and suggests that motivation factors will continue to change as the discipline evolves, necessitating further research.

Hall *et al.* (2009) analysed the use of theory in studies investigating the motivations of software engineers using the results from their systematic literature review (Beecham *et al.* 2008). Of the reviewed studies ($n = 92$) classic theory¹ was only explicitly used in 51 studies. Hall *et al.* (2009) found that only 33 studies use classic theory as a basis for the study and that the intention was to replicate, validate or extend classic theory. This use of classic theory is described as “the most substantial use of classic theory” Hall *et al.* (2009).

¹ The eight classic motivation theories are Equity theory, Stimulus response theory, Job Characteristics theory, Goal Setting theory, Expectancy theory, Maslow’s Need theory, McClelland’s Need theory, and Hygiene theory.

With only 51 of the reviewed studies using theory and only 33 of those using it in the most substantial way, it is possible that a lot of the previous research does not sufficiently consider motivation theory when investigating the motivations of software engineers.

This paper presents the data from recent interviews with software engineers that explored their perceptions of factors in motivation.

2. Research Method

Motivation is a collection of internal processes (Baron 1991) which makes it difficult to observe directly. Investigating motivation requires the elicitation of state of mind information from participants. We used semi-structured interviews in order to engage participants with the subject while eliciting their own experiences, reflections and stories in their own terms.

One-to-one interviews with the participants at their workplace were used to gather data. All of the interviews followed a framework of initial questions. The questions were open so that they did not lead participants towards specific responses. Interviews were audio recorded, and field notes were taken. A pilot study had been conducted previously to evaluate the protocol. The interviews lasted as long as it took for the participants to answer all questions, there was no time limit and the duration of each interview ranged from 24 minutes to 72 minutes.

Semi-structured interviews were used to allow the participants to discuss the questions and their responses with fewer constraints and to enable the researcher to probe further into the participants' responses when required. The specific questions used in this study aimed to explore the participants' motivation by getting them to talk about work they enjoyed and didn't enjoy as much while avoiding the use of terminology from the motivation literature.

The participants were asked to complete a computer-based personality inventory after the interviews, which they all completed within two weeks.

The personality inventory was composed of questions from the International Personality Item Pool (IPIP) proxy for Costa and McCrae's NEO-PI-R domains (Goldberg *et al.* 2006). The same questions have been administered in the same order to over 3.6 million people via the myPersonality project (myPersonality Database 2011). This personality inventory was selected due to its good accessibility, available data for comparison, and being widely used in the academic community.

The personality inventory serves a dual purpose by enabling a comparison between the participants from this study with other software engineers, and in future analysis allowing for an investigation between personality characteristics of the participants and other data collected during the study.

2.1. Interview Questions

The initial questions used for each interview were divided into three topics: demographic information, motivation factors, and feedback received by software engineers. The researcher also posed follow-up questions for elaboration or clarification.

The initial questions concerning motivation factors are shown below in the order in which they were asked:

1. On any of your recent work – tell me what you enjoyed about it?
2. What encourages you to go that 'extra mile' at work?
3. What about any of your recent work that you didn't enjoy as much?
4. Is there something that really saps your energy at work?
5. Is there a part of being a software engineer that you prefer doing over other aspects?
6. What about other aspects of your role, what do you like doing least?
7. So it's Wednesday morning, middle of the week. You've just woken up, what makes you get up and go to work as a software engineer?

This paper will focus on the demographic information, the motivation factors, and the personality inventory results.

2. Participant Profile

In this section the overall profile of the participants is presented. This includes a description of their workplace, their demographics details, and their scores from a personality inventory.

2.1. Participant Workplace

The participants are all full-time employees of the Red Gate software company. Red Gate is based in Cambridge and employs around 200 people, including 29 software engineers. Red Gate creates “ingeniously simple” products and specializes in MS SQL Server, Oracle, .NET and email archiving tools.

2.2. Demographics

The experience ranged from 6 months to 24 years for the participants. Each participant had been at Red Gate between 6 weeks and 5 years. All of the participants were male. All of the participants have a degree, and 12 of those 13 degrees were in computer science or a closely related subject.

The participants were all members of 6 different teams within Red Gate that adopted an agile approach to software development, although the level of adoption varied between teams. Most of the developers were typically part of small development teams of 3-6 developers.

2.3. Personality Inventory

Figure 1 displays the mean scores and standard deviation for the 13 participants and for 130 people who completed the myPersonality assessment and stated their job title as “software engineer”. The myPersonality data was filtered so that only participants who had completed the same 100 questions and who had also only attempted to complete the personality assessment twice were included in this analysis. This was done to ensure that the data was suitable to be compared and to exclude respondents who may have tried to idealise their results.

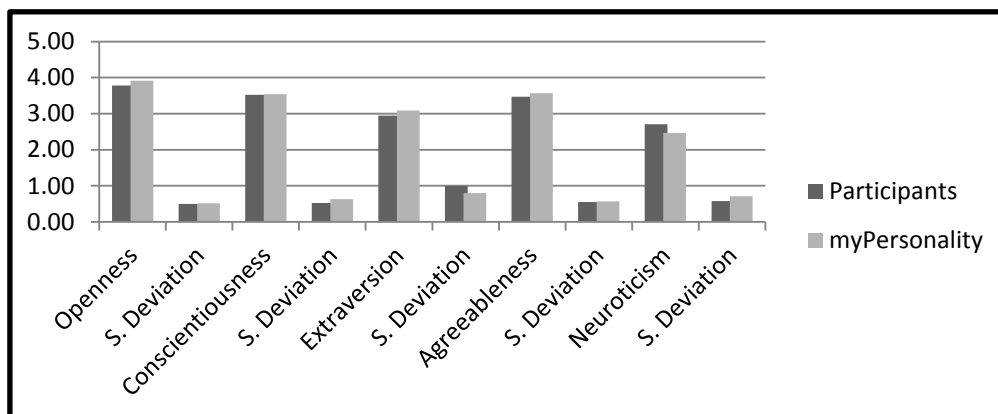


Figure 1. Personality Comparison Chart

Table 1. Personality Comparison Chart Statistics

	Openness	S. Dev	Conscientiousness	S. Dev	Extraversion	S. Dev	Agreeableness	S. Dev	Neuroticism	S. Dev
Participants	3.78	0.50	3.52	0.53	2.95	0.99	3.47	0.55	2.71	0.57
myPersonality	3.91	0.52	3.54	0.63	3.09	0.80	3.57	0.57	2.47	0.71

From this comparison we can conclude that the 13 participants are representative of the software engineering population.

3. Results

This section describes the analysis of the interviews briefly and presents the results.

The audio recordings of the interviews were transcribed and the analysis was conducted question-by-question. The analysis per question was inductive, identifying emergent themes related to motivation strictly on the bases of what was evident in the data.

The number of individual participants who discussed each theme in response to each question are presented in the following sub-sections. Some participants raised the same theme multiple times per question, but the data presented focuses on themes from each participant and not the occurrences of each theme.

Some of the responses given during the interviews did not directly answer the question, and these were excluded from the final analysis.

3.1. Motivation Factors

This section presents highlights from the analysis of the responses to each of the motivation factor questions in turn.

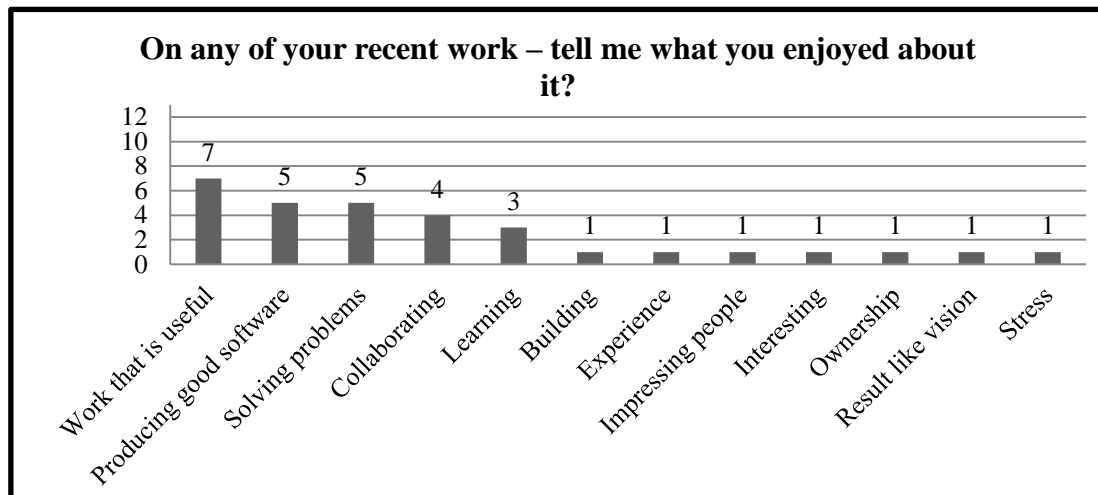


Figure 2. Question 1 Themes

Question 1 (Figure 2) on any of your recent work – tell me what you enjoyed about it: ‘Work that is useful’ was the most frequent theme to emerge from responses of 7 of the 13 participants, with responses including “most rewarding things that I’ve found from software have been seeing people use what I’ve written and finding it useful solving problems, so when both of those happen it’s amazing” and “seeing something that you’ve built be part of something that’s just solved someone’s problem and make their world a better place and save them some time or cash or whatever, I think that’s the, that’s the really cool thing. You know it’s to see other people go wow”.

Another frequent theme (5/13) was producing good software, and this included improving on previous software; “removing or at least changing this horrible code and putting something nice and shiny in instead”. Solving problems was a third frequent theme (5/13), e.g.: “Solving the problems is good fun, so finding nice solutions, discussing them, realising they’re [not so good] and finding better solutions”.

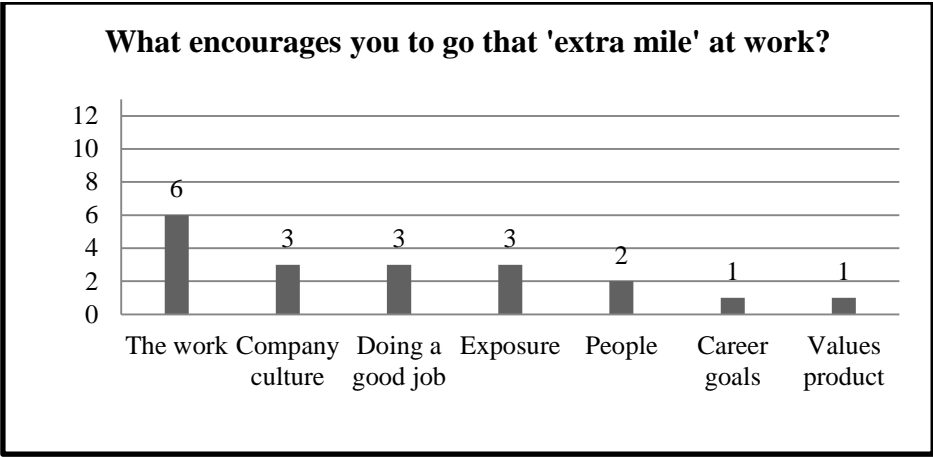


Figure 3. Question 2 Themes

Question 2 (Figure 3) what encourages you to go that 'extra mile' at work: 'The work' was discussed by 6 of the 13 participants, e.g.: "I think it's just the combination of enjoying writing the code knowing that what you're producing is going to be cool" and "it's that thought that we're going to produce this product to a good standard and we're going to get it out when we said we're going to get it out."

'Company culture' was illustrated by one participant stating "I'm still reasonably inclined if not more so to actually put in the effort here where it's needed..." "...because of the way that Red Gate generally operates in that there is a lot of flexibility, I know that for example if it's slightly more quiet time of the project, or even if it isn't, if I need to go out in the afternoon for a house viewing or something like that then there's not a problem."

'Doing a good job' was reported by three participants, with responses including "I never like to feel like I've done a half arsed job of anything" and "I wouldn't be happy with just writing hacky code quickly. I want to do it correct[ly]."

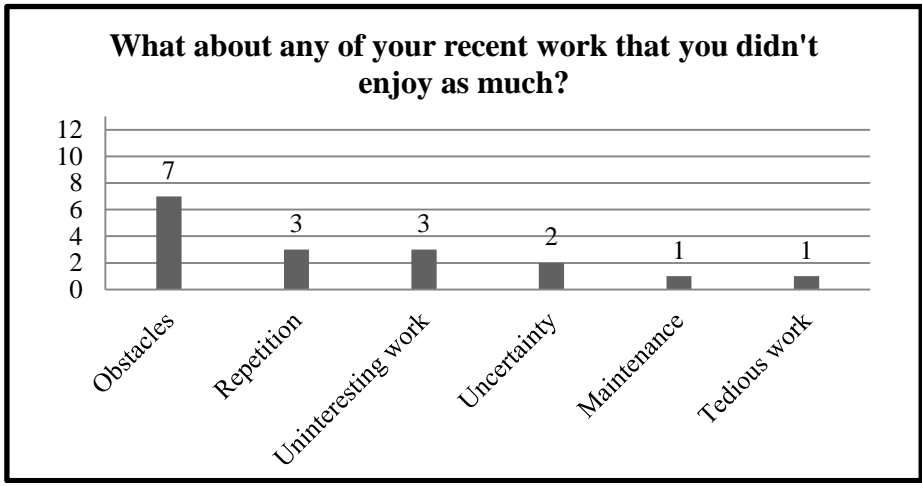


Figure 4. Question 3 Themes

Question 3 (Figure 4) what about any of your recent work that you didn't enjoy as much: 'Obstacles' (7/13) was a frequent theme. One participant encapsulated obstacles concisely with "your job as a software engineer is to write code and to write good code and things that make it hard are where other things get in the way", and another participant stated "anywhere where again there are outside obstacles or be it systems that are not directly under your control which are operating badly or worked and you didn't touch anything and then they stopped working." 'Obstacles' emerged as a theme from participants responding with different obstacles that got in the way of their work.

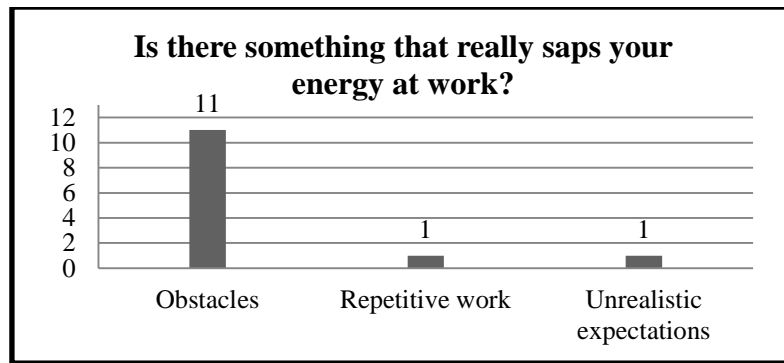


Figure 5. Question 4 Themes

Question 4 (Figure 5) is there something that really saps your energy at work: ‘Obstacles’ was a frequent theme in response to question 4. ‘Obstacles’ included a range of factors including a poor code base “when you’ve got a large complicated poorly written code base and you have to add some small feature to it and you think well this shouldn’t take very much time and it’s just taking forever”, lack of project direction “not really knowing where you should be going and no one setting it is always a challenge” and complicated contingencies “like you fix something, and something else over here breaks so you fix that, and something else over here breaks so you fix that and it just keeps on going round and it’s horrendous.”

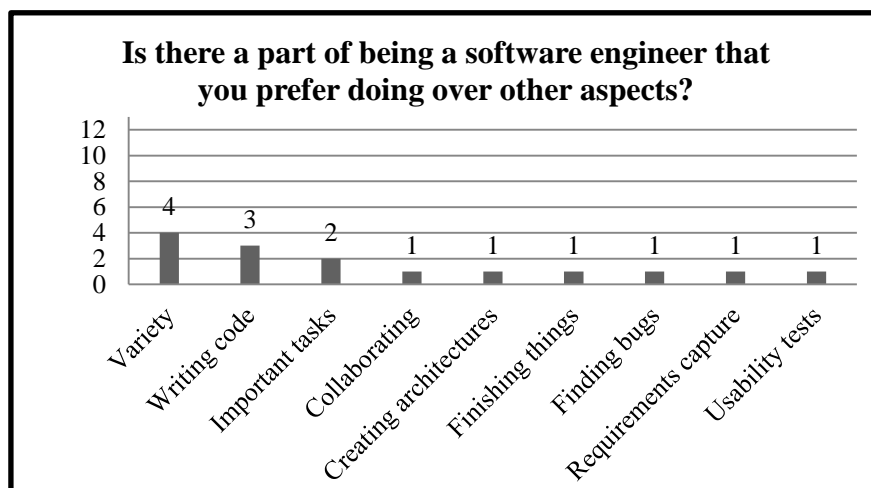


Figure 6. Question 5 Themes

Question 5 (Figure 6) is there part of being a software engineer that you prefer doing over other aspects: As illustrated in Figure 7 there was a spread of responses. Some participants (4/13) enjoyed the ‘variety’, explaining “I have a variety of quite different things that I enjoy doing” and “I enjoy that I write code at all levels of the stack and that I do the testing and the development and I go off and talk to sort of essentially people who care about the product and get the requirements.”

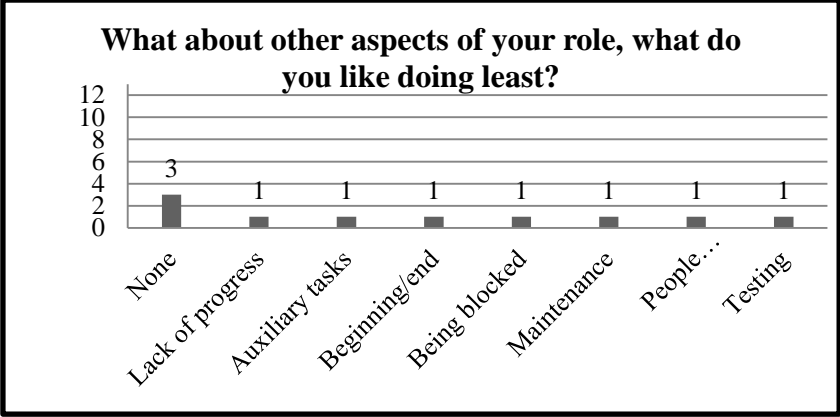


Figure 7. Question 6 Themes

Question 6 (Figure 7) what about other aspects of your role, what do you like doing least: While a few participants (3/13) stated that they did not have a non-preference “there's no particular bit that I dislike”, and some of the participants did not answer the question, each of the others gave a different response. The participants appeared to struggle to answer question 5 and question 6, and they were suitably deemphasised as further investigation is required.

Question 7 (Figure 8) so it's Wednesday morning, middle of the week. You've just woken up, what makes you get up and go to work as a software engineer: The participants provided a range of reasons that get them up in the morning. Seven participants said they had work to continue “it's because I've left something in an incomplete point, I quite want to finish it” and “Typically because on Tuesday you left something in that isn't finished yet.”

Six participants discussed making something; “I get to create something that's useful and it's used by lots of people to help get their jobs done” and “when I go to work I've made something by the end of the day.”

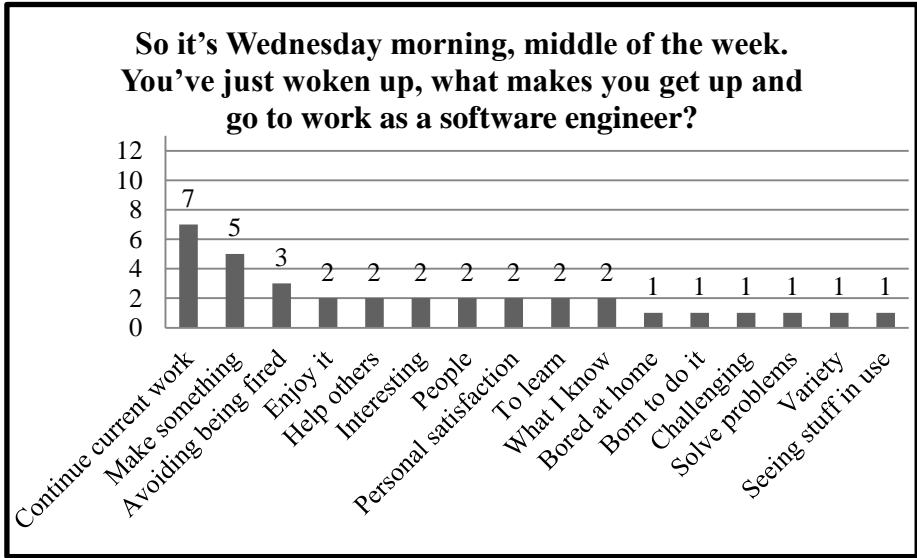


Figure 8. Question 7 Themes

4. DISCUSSION

‘The work’ is the most frequent theme emerging from responses to the questions investigating the enjoyable and potentially motivating areas of software engineering. It appears in responses to three of the four questions (question 1, 2 and 7). ‘The work’ is listed in Herzberg’s Hygiene theory (1987) as the third strongest *motivator* (after achievements and recognition). The data presented here has similar

findings on the importance of ‘the work’ to Herzberg’s research which was first published over 50 years ago.

‘The work’ corresponds to the most commonly reported motivator in the original systematic literature review, “technically challenging work” (Beecham *et al.* 2008), although not exactly, as participants in this study also discuss ‘the work’ as being interesting, useful, and enjoyable which is not explicitly technically challenging work.

‘Obstacles’ is the most frequent theme emerging from responses to the questions investigating the less enjoyable and potentially de-motivating areas of software engineering. It appears in responses to questions 3, 4 and 6, and encompasses a range of different factors including being disrupted during work, being held back by other team members, lack of direction during a project, and dealing with poorly written code. ‘Obstacles’ do not appear in any of the classic motivation theories, but this is not unexpected as the theories mainly discuss motivating factors and ‘obstacles’ as discussed by the participants is not a motivator.

‘Obstacles’ emerged as a theme from these different factors because the participants discussed them as things that get in the way of the work. ‘Obstacles’ *per se* do not appear in the systematic literature review by Beecham *et al.* (2008) or the systematic literature review update (Franca *et al.* 2011), but some factors that may be considered ‘obstacles’ do, for example poor communication and poor management. ‘Obstacles’ hamper the motivational potential of ‘the work’ and sap the energy of software engineers.

‘People’ were identified as a frequent theme discussed by 8 of the 13 participants across all the questions. Previous work (Sach *et al.* 2010, Sharp and Hall 2009) also identified ‘people’ as a common motivator, 13 of the 15 participants in the Sach *et al.* (2010) study included ‘people’ in their responses. ‘Interpersonal relationships’ is the second strongest *hygiene factor* in Herzberg’s Hygiene theory (1987). Hygiene factors are considered to lead to dissatisfaction if not present and a neutral feeling if they are present. The data presented here suggests that ‘people’ may have a different influence on motivation because ‘people’ are discussed in response to questions investigating the enjoyable and potentially motivating areas of software engineering.

The higher incidence of ‘people’ in the Sach *et al.* (2010) study could be because that data was collected from self-selecting participants attending a conference whereas the data in this study was collected in-situ with software engineers at their own working environment.

This study focused on a small group of software engineers working in a small to medium sized company with around 200 employees and under 30 software engineers. The data collected here may be different to data collected from participants at a larger software engineering company.

4.1. Limitations

The personality scores recorded from the participants show them to be representative of the wider software engineer community, but as they all work in the same organisation this might influence their perceptions.

Given the low number of participants and the participation of software engineers from the same company, the results can only be taken as indicative of software engineers.

4.2. Future Work

A prominent future research direction would be to repeat the study with a different group of software engineers from a different environment or a range of environments. This would aid in investigating the impact of environment on perceptions of motivation.

‘The Work’ is the most commonly reported motivation factor in the literature (Beecham *et al.* 2008), but is described as ‘technically challenging work’. The data presented in this paper discusses the work as interesting, useful, and enjoyable which is different to the literature. Future research should investigate the difference in how ‘the work’ is discussed by participants.

'Obstacles' is not present in the reviewed software engineer and motivation literature, suggesting that it may be an emerging factor in the motivation of software engineers. With 11 of the 13 participants discussing 'obstacles' in response to "is there something that really saps your energy at work?", this theme requires further investigation in order to establish whether 'obstacles' are present in other contexts, and if so what impact they may have on motivation or de-motivation.

Another area requiring further investigation is identifying the importance of 'people' factors for software engineers. The focus of this research could be on the interaction between software engineers and specifically the feedback that occurs in software engineering environments.

4. ACKNOWLEDGEMENTS

Thank you to all the participants from Red Gate in Cambridge for their time.

5. REFERENCES

- Baron, R. (1991) 'Motivation in work settings: Reflections on the core of organizational research', *Motivation and Emotion*, Vol.15, Issue 1, pp1-8.
- Beecham, S., Baddoo, N., Hall, T., Robinson, H. And Sharp, H. (2008) 'Motivation in Software Engineering: A Systematic Literature Review', *Information and Software Technology* 50, no. 9-10 (August 2008): 860-878.
- Boehm, B.W. (1981) 'Software Engineering Economics', Englewood Cliffs: Prentice Hall.
- DeMarco, T. and Lister, T. (1999) 'Peopleware – Productive Projects And Teams', Dorset House.
- França, A. C. C., Gouveia, T.B., Santos, P.C.F., Santana, C.A., Da Silva, F.Q.B. (2011) 'Motivation in Software Engineering: A Systematic Literature Review Update', *Proceedings of the 15th International Conference on Evaluation and Assessment in Software Engineering (EASE '2011)*.
- Goldberg, L. R., Johnson, J. A., Eber, H. W., Hogan, R., Ashton, M. C., Cloninger, C. R., and Gough, H. C. (2006) 'The International Personality Item Pool and the future of public-domain personality measures. *Journal of Research in Personality*', 40, pp84-96.
- Hackman, J. R. and Oldham, G. R. (1976). 'Motivation Through the Design of Work: Test of a Theory', Academic Press, New York.
- Hall, T., Baddoo N., Beecham, S., Robinson, H., and Sharp, H. (2009) 'A Systematic Review of Theory Use in Studies Investigating the Motivations of Software Engineers', *ACM Transactions on Software Engineering and Methodology*, vol. 18, issue 3, pp1-29.
- Hall, T., Sharp, H., Beecham, S., Baddoo N., and Robinson, H. (2008) 'What Do We Know about Developer Motivation?', *IEEE Software*, vol. 2008, pp. 92-94.
- Herzberg, F.I. (1987) "One more time: How do you motivate employees?", *Harvard Business Review*, vol. 65 issue 5, pp109-120.
- myPersonality database. [online] Available at: <<http://www.psychometrics.ppsis.cam.ac.uk/page/255/mypersonality-database.htm>> [Accessed 10 June 2011].
- Procaccino, J. D., Verner, J. M., Shelfer, K. M., and Gefen, D. (2005) 'What do software practitioners really think about project success: an exploratory study', *Journal of Systems and Software*, vol. 78, issue 2, pp194-203.
- Sach, R., Sharp, H., and Petre, M. (2010) 'Continued Involvement in Software Development: Motivational Factors', *The 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 16-17 Sep 2010, Bolzano, Italy.
- Sharp, H. and Hall, T. (2009) 'An initial investigation of software practitioners' motivation', *Cooperative and Human Aspects on Software Engineering, CHASE '09, ICSE Workshop*, pp84-91.