ISSN 1744-1986

The Open University

Technical Report N° 2007/13

Exploring Motivational Differences between Software Developers and Project Managers

Helen Sharp
Tracy Hall
Nathan Baddoo
Sarah Beecham

27th September 2007

Department of Computing
Faculty of Mathematics and Computing
The Open University
Walton Hall,
Milton Keynes
MK7 6AA
United Kingdom

http://computing.open.ac.uk

Exploring Motivational Differences between Software Developers and Project Managers

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

Tracy Hall
School of Computer Science
University of Hertfordshire
Hatfield, AL10 9AB
United Kingdom
t.hall@herts.ac.uk

Nathan Baddoo
School of Computer Science
University of Hertfordshire
Hatfield, AL10 9AB
United Kingdom
n.baddoo@herts.ac.uk

Sarah Beecham
School of Computer Science
University of Hertfordshire
Hatfield, AL10 9AB
United Kingdom
s.beecham@herts.ac.uk

ABSTRACT

In this paper, we describe our investigation of the motivational differences between project managers and developers. Motivation has been found to be a central factor in successful software projects. However the motivation of software engineers is generally poorly understood and previous work done in the area is thought to be largely out-of-date. We present data collected from 6 software developers and 4 project managers at a workshop we organized at the XP2006 international conference. We collected this data using the Repertory Grid Technique (RGT). RGT originated from psycho-analysis and allows researchers to uncover the detailed building blocks of peoples' attitudes. In this investigation we elicit RGT data focused on attitudes to motivation. We compare the motivation attitudes of software developers to project managers. Our findings suggest that project managers and software developers think differently about motivation. It is very important for successful project outcomes that project managers understand that developers may be motivated differently to themselves and that they manage developers' motivations appropriately.

Categories and Subject Descriptors

D.2.9 [Management] *Productivity*, *Programming teams*

General Terms

Human Factors, Management

Keywords

Motivation, Repertory grid technique

1. INTRODUCTION

Motivation in Software Engineering is reported to have the single largest impact on practitioner productivity (Boehm, 1981) and software quality management (McConnell, 1996), and continues to be 'undermined' and problematic to manage (Procaccino et al, 2005). Motivation is increasingly cited as a particularly pernicious people problem in Software Engineering. In DeMarco and Lister's (1999) survey, motivation was found to be one of the most frequently cited causes of software development project failure.

The Standish report (1995) amplifies this finding by reporting that having access to competent, hard working and focused staff is one of ten success criteria for software projects. It is therefore important that project managers have a good understanding of software developers' motivators. Such understanding should enable project managers to manage developers' motivators so that the quality and quantity of development work is improved. This improvement could significantly affect successful project outcomes.

Some studies suggest that conventional approaches to motivation in software engineering might be outdated. Previous approaches have concentrated on rewards and recognition, e.g. ProjectLink (2006). However Software Engineers have been identified as having a distinctive personality profile (Capretz, 2003) that are instead motivated by the nature of the job, e.g. technical success, challenging technical problems (Tanner, 2003; Ramachandran and Rao, 2006) and peer interaction (Klenke and Kievit 1992; Linberg 1999; Andersen 2002; Tanner 2003; Procaccino et al. 2005). In our systematic review of the literature on motivation in software engineering (Beecham et al, 2007) we found an increasing awareness of the importance of motivating software developers. We also found that motivation is context-sensitive, i.e. whether or not an individual is motivated depends on their own circumstances, such as career stage, personal abilities and their job role.

In this paper, we investigate how motivation varies between 2 specific practitioner roles: project managers and developers. We present detailed data on the motivators of 6 developers and 4 project managers. We collected this data during a workshop that we ran at the XP2006 International Conference at Oulu, Finland. We targeted this conference because agile approaches have been found to improve job satisfaction (Syed-Abdullah et al. 2005).

We collected data on motivation from developers and project managers using the Repertory Grid Technique (RGT) (Fransella and Bannister 1977). This is a well established technique that originated in psycho-analysis but has subsequently been widely used in a range of other disciplines. This technique uncovers the basic building blocks of the attitudes people have developed to particular issues. It is a particularly relevant technique for exploring differences between peoples' attitudes. Consequently it is an appropriate technique for identifying developers' and project managers' attitudes to motivation.

The rest of the paper is organized as follows: In Section 2 we present some background to motivation in software engineering and go on to discuss motivation in terms of agile development approaches. In Section 3 we describe RGT and show how we used this research methodology. We summarise our findings in Section 4 and discuss them further in Section 5. Finally we conclude in Section 6.

2. PREVIOUS WORK ON MOTIVATION

2.1 Motivation of Software Engineers

Motivation refers to the initiation, direction, intensity and persistence of behaviour. Motivation, although very important is a soft factor, and difficult to quantify, consequently managing it often takes a backseat (McConnell 1998). Our previous work reveals motivation as a complex phenomenon with many inter-related, context-dependent factors (Beecham et al. 2007). The Software Engineering workforce is organised according to a variety of environmental factors and differing software practitioner roles that are time dependent and culture-specific (Orlikowski and Baroudi 1989). The positive influence motivated engineers have on the project is recognised in the Agile manifesto, that states "build projects around motivated individuals" http://www.agilemanifesto.org.

Although we have found that more work has been published in the area of motivation in software engineering over the past few years (Beecham et al. 2007), few studies recognise the need to clearly differentiate between different roles. Prasad et al (2005) and Kaarst-Brown & Guzman (2005) report that in the software engineering literature it is normal to treat all practitioners as a homogeneous group who are all similarly motivated. Yet the literature shows that "one size does not fit all"; indeed many researchers call for a more defined approach to managing practitioners in their different roles (Kaarst-Brown and Guzman 2005; Prasad et al. 2005). The need to look beyond stereotypes is also a central message in (Enns et al. 2006) who found that IT practitioners "possess a diversity of motivations that cut across age and organizational tenure profiles". This group of researchers suggest that rather than rely on simple generalizations, managers must evaluate an intricate combination of motives (such as achievement, security, and flexibility) in conjunction with career stages to understand the needs of individual software developers. Managers should look beyond stereotypes and strive for a richer understanding of software developers.

Different roles in software engineering attract different types of personality, for example (Jekielek 2002) found that control systems professionals are often introverts who tend to prefer working with objects rather than with people. In our previous study of the problems developers, project managers and senior managers have with software process improvement (SPI), we found that practitioners' problem priorities differ to reflect their varied experiences and roles (Beecham et al. 2003). We found in a subsequent study, where we examined differences in de-motivators for SPI across staff groups, that these differences are associated to the role that software practitioners have in software engineering (Baddoo and Hall 2003).

The importance of recognising roles in software engineering is not new. Whitaker (1997) looked at different roles in motivating software engineers. Goldstein and Rockart (1984) extended Hackman and Oldham's model of motivation (JCT) (Hackman et al. 1975) in the 1980s to include roles.

2.2 Agile and Motivation

The data we analyse in this paper was collected from developers and project managers attending an agile international conference (XP2006). There is evidence to suggest that Agile approaches are particularly relevant to improved motivation and job satisfaction. Syed-Abdullah et al.'s (2005) longitudinal study revealed that eXtreme Programming (XP) has a positive impact on an individual's disposition to be happy, across time and situations. This finding is supported by Mannaro et al's (2004) study, where job satisfaction was found to be higher in developers using XP practices as opposed to developers not using XP practices. Practitioners also showed a strong preference for working in an XP environment, using XP practices.

Melnik and Maurer (2006) empirically compared job satisfaction in Agile and Non-Agile Software development teams finding that the greater the experience of working in an Agile environment the greater the job satisfaction. In particular Melnik and Maurer (2006) report "…increased individual team morale, motivation, performance productivity and retention." Asproni (2004) explains how Agile development methods contain the necessary ingredients to motivate developers to make effective teamwork possible. Law and Charron (2005) found pair programming motivating in two separate projects because it addressed the need for learning, autonomy and social activity. However, Law and Charron (2005) do accept that pairing can have a negative impact if pairs have personality conflicts.

Initial findings from our recent empirical study that examined whether the XP environment meets the motivational needs of the software developers (Beecham et al, 2007) indicate that XP provides supportive conditions for nurturing and motivating software developers to work well in a team, to create high quality software and to increase productivity. On the other hand, our observational data revealed the XP environment to be potentially de-motivating as it does not support the need for individual recognition, for clear career progression, and variety of work. Furthermore, pair programming may weaken the developer's ability and confidence to work alone, should the developer need to move out of an XP environment.

3. RESEARCH METHODS

3.1 Repertory Grid Technique

We use the repertory grid technique (RGT) (Fransella and Bannister 1977) to collect and analyse software practitioners' perception of motivation. RGT was initially designed for use in psycho-analysis but has subsequently been generalised for broader use. It has been successfully used in many fields, eg, education and market research. Indeed in software engineering, RGT has been used as a method for eliciting software requirements (Maiden and Rugg 1996). RGT is an investigative data collection technique which attempts to eliminate the influence of the researcher's own perspective on participants. The aim of RGT is to allow participants to reveal their beliefs without any influence from the researcher. RGT enables researchers to understand how participants have developed their perspective on a given situation. It enables the researcher to uncover the "building blocks" of an opinion. In this study we attempt to

uncover individual's attitudes towards software engineering with regards to their own motivation.

Although RGT is typically used to elicit perceptions from individuals, in this study we have applied RGT to small groups, pairs of practitioners and individuals. RGT used with groups has not been found to compromise the data collected (Shaw and Gaines 1996; Thomas and Harri-Augstein 1985).

The main aspects of RGT are:

Elements. These are the subject of RGT analysis. Stewart et al describe elements as "people, objects, events and activities" (Stewart *et al.* 1981). In RGT studies it is not uncommon for the participant in the study to be part of the set of elements in the study. Elements are always grouped together in three's (triads). This is to enable detailed consideration about each element in the context of how it is similar to and different from the other elements in the triad. In this study we use elements in two different ways. First, we ask practitioners to propose their own triad of motivation elements, second we ask practitioners to consider specific motivation elements in triads we have formulated. These specific motivation elements are based on those identified as important in the motivation literature (these are presented in Table 1).

Constructs. Stewart et al describe an individual's construct system as their hypotheses by which they interpret the world. These interpretations emerge from their experiences of the world (Stewart *et al.* 1981). Constructs are based on specific elements and can be explained as a perception of a given element. For example, when considering the element *ownership of work* a developer may say that this encourages good work - *"encourages good work"* then becomes a construct of the element *ownership of work.* Using RGT constructs are always elicited as bi-polar constructs.

Bi-polar constructs. RGT aims to elicit from participants constructs that distinguish between elements. Bi-polar constructs are the core feature of RGT. Eliciting bi-polar constructs of element triads enables rich analysis of complex attitudes to related issues. Participants are asked to consider triads of elements in 2-to-1 groupings. They are asked to identify how the 2 elements in the group are similar to each other but different from the single element. For example in considering the elements: *ownership*, *responsibility* and *working with others*, the first two elements will be grouped together in opposition to the third element. In considering how *ownership* and *responsibility* are similar to each other but different from *working with others*, developers may say that ownership and responsibility *enables work* whereas working with people *directs work*. In this example the bi-polar constructs are "enables work", and "directs work", as these distinguish between ownership and responsibility on one hand and working with others on the other. There are 3 such groupings within each triad which results in rich data being collected on how each element is perceived.

Grid analysis. As data is collected from participants using RGT, the relationship between elements and constructs that participants identify is recorded on a grid. A grid of constructs and elements is the basis of RGT data analysis. The grid contains alignment positions that participants give in relation to the extremes of a bi-polar construct. This alignment position represents their own perceptions of where on the bi-polar continuum their attitudes are. This grid is then used to understand participants' attitudes to issues of interest. For example once the bi-polar constructs for *ownership of work* have been listed in a grid, an issue of interest might be *your current work tasks*. Participants would then mark the place on the gird, between the bi-polar constructs to represent where on the grid the issue of interest aligns to the extremes of the bi-polar constructs.

3.2 An example of RGT

3.2.1 Identifying elements

A triad of elements must be identified. Elements can be elicited from participants but in this example the following pre-set elements form the example triad:

Example Triad

E1: An agile guru
E2: A plan-driven expert
E3: An Open Source project leader

3.2.2 Eliciting bi-polar constructs

Bi-polar constructs are elicited from participants in 2-to-1 combinations of elements in the triad. In this example the following questions would elicit bi-polar constructs:

1. What is similar about an *agile guru* and a *plan-driven expert* that makes them different from an *open source project leader*?

   Answers to this question result in the E1&E2vE3 part of the grid.

2. What is similar about an *agile guru* and an *open source project leader* that makes them different from a *plan-driven expert*?

   Answers to this question result in the E1&E3vE2 part of the grid.

3. What is similar about an *open source project leader* and a *plan-driven expert* that makes them different from an *agile guru*?

   Answers to this question result in the E2&E3vE1 part of the grid.

The following bi-polar grid might emerge for this example:

Part of grid	**Participant generated bi-polar constructs**	
E1&E2vE3	Employed by a company……....	Doing voluntary work
	Developing commercial Software……..…….…	Developing free software
E1&E3vE2	Dynamic approach …….……	Conventional approach
	Exciting ……..	Boring
	Young……....	Old
E2&E3vE1	Produce too much documentation……..	No documentation
	Conventional methods used…….	Innovative methods used

This example grid shows how in eliciting bi-polar construct opinions, beliefs and mis-conceptions come to light when

participants are asked to consider similar issues in an RGT exercise.

3.2.3 Grid analysis

The final stage in RGT is to ask participants to apply their bi-polar constructs to issues of interest. This is done by asking people to identify positions on the grid that represent alignment of the issue of interest to the bi-polar constructs. The following grid provides an example of analyzing:

'*Where does your current project rate on the grid*?'

Bi-polar extreme	Alignment							Bi-polar extreme
Employed by a company	1 ✓	2	3	4	5	6	7	Doing voluntary work
Developing commercial software	1	2	3 ✓	4	5	6	7	Developing free software
Young	1	2	3	4	5	6 ✓	7	Old
Dynamic approach	1	2	3	4	5	6 ✓	7	Conventional approach
Exciting	1	2	3	4	5 ✓	6	7	Boring
Produce too much documentation	1	2 ✓	3	4	5	6	7	No documentation
Conventional methods used	1	2 ✓	3	4	5	6	7	Innovative methods used

This example grid illustrates how RGT can be used to identify the important constructs that people use to form their attitudes. These constructs are then used as the basis of understanding their current attitudes about a very specific issue. RGT has proved to be a very powerful tool in gaining a rich understanding of peoples' thinking in a variety of domains.

3.3 Our Implementation of RGT

3.3.1 Overview

We collected data for this RGT study during a 90 minute workshop at the XP2006 conference held in Oulu, Finland (Sharp et al 2006). Twenty conference attendees self-selected to take part in the workshop. On entering the workshop room, attendees were asked to group themselves according to 4 categories, based on practitioner roles: developer, project manager, mentor/coach and customer. Participants were given a brief overview of the RGT method and some worked examples of the method were presented. Participants then worked through several RGT exercises as described in the rest of this section.

3.3.2 Materials

Each participant was provided with a set of data capture forms. This set of forms consisted of one form asking for demographic information, one form relating to some general motivation questions, one form to capture participants' own triad of motivation elements and one form to document the grid constructs built around three pre-set triads of elements.

3.3.3 Data collection

The workshop was structured around collecting RGT data. There were 4 main sections in the workshop each of which concluded with a plenary discussion. After a brief introduction, the first section entailed a general discussion on motivation in role groups in order to set the scene, and to orient everyone's thinking towards motivational issues. This discussion in role groups was prompted by the following question: "What aspects of your job do you get most satisfaction from?" Groups discussed their answers among themselves and then in a plenary discussion.

In the second section attendees were asked to brainstorm within their role groups and identify RGT elements that they felt were important in the context of motivation. Each individual was then asked to document their own choice of three elements to make a triad. These were recorded on the workshop forms.

In the third section of the workshop, we asked participants to consider the three triads, shown in Table 1, that we identified as important motivational issues from the literature on motivation.

Table 1. Pre-set triads

Triad 1 elements	Triad 2 elements	Triad 3 elements
Technical Challenge	Working with people	Autonomy
Good tools	Recognition	Job security
Rewards and benefits	Ownership	Career prospects

Working in pairs within their role groups, they were then asked to develop bi-polar constructs for these triads.

Finally, in the fourth section participants constructed a grid based on the bi-polar constructs they had recorded on their forms. Individuals were asked to align on a scale 1-7, all constructs according to:

A. Where does agile rate?

B. Where does plan-driven rate?

C. Where do you rate?

3.3.4 Data analysis

The results were themed using a categorisation scheme that emerged from the data itself. Two separate categorisation schemes were identified: one for the elements and one for the constructs. The elements categories are listed below; the construct categories are defined in Table 2.

Element category
Problem-solving
Success
Learning
Creativity
Users
Influence
People
Making a difference
Clear presentation

There was a great deal of commonality in the elements that practitioners generated themselves in that many common words were used to label elements. Consequently elements were

categorized very simply according to the words used. For example 6 practitioners generated elements they labeled as 'problem-solving', consequently a *problem-solving* category was used. Direct mappings from the labels practitioners used for elements, to classifications was used to categorize all elements into one of the above classifications.

We classified all constructs generated in response to the 3 element triads as shown in Table 2.

Table 2. Definitions of construct themes

Construct theme	Definition
Part of job	Relating to intrinsic aspects of doing actual work and the way the work is done or the process and tasks undertaken.
Technical Issue	Whether an issue is directly referred to as technical or not
Time-based issue	Reference to temporal aspects, referring to time in terms of now, future or always. Or change over time. Or temporal aspects of process.
Intangible	Direct reference to the tangibility or abstract nature of issues.
Personal or team	Reference to people issues in terms of individuals or teams or the organisation. Also includes reference to management and independence of work.
Motivation	Reference to motivation in terms of satisfaction, stimulation, excitement or making work easier or work environment better. Includes reference to generally feeling positive about work and being able to work effectively
Productivity	Direct references to work productivity
Creativity	Direct references to creativity
Positive or negative	Direct reference to positive feelings about an issue. Includes use of terms like good and easy as well as how significant an issue is.
Misc	Issues not classified in above categories

Both classification themes were generated using a grounded approach. All the individual elements and constructs that participants generated were grouped together. Themes were then identified for these groupings. For example the following raw bi-polar constructs formed a sub-set of the *time-based* theme:

Raw bi-polar constructs in the time-based theme	
needed every day	in between
during the process	can come after
effect outcome	can come after
needed every day	can be less frequent
current	retrospective

In both cases, the scheme was developed and defined iteratively from the data collected. Our classification scheme has the 'grounded flavour' described by Robson (2002). The scheme was developed by one researcher, and applied to the data. An inter rater reliability test was then performed by a second researcher who categorised the data independently. This led to an initial agreement of 100% for the element categories and 82% for the construct categories.

4. FINDINGS

4.1 Participants

In this study we are particularly interested in practitioner motivation. Consequently we removed data collected from researchers and educators. The remaining practitioner data came from 6 developers, 4 project managers, 2 mentor/coaches and 1 customer (see Table 3). Because of the small number of mentor/coaches and customers we do consider them further. We therefore focus on data from 4 developers and 6 project managers. All data we present for developers and project mangers has been normalized to account for the imbalance in the representation of each of the two groups. Consequently results between developers and project managers can be compared directly despite the fact that there were more project managers than developers.

Table 3 shows that half of the software developers have significant software development experience. However there is lower that expected experience of using agile approaches (considering the data was collected from participants at an agile conference). There is a more consistently high level of experience of software development amongst the project managers though their experience of agile is mixed.

Table 3a. Practitioner background

Nationality	Agile experience (years)	Software development experience (years)
Developers		
Dutch	1	8
Estonian	< 1	2
Finnish	0	3
Finnish	< 1	3
Finnish	0	12
Finnish	0	8

Table 3b. Project manager background

Project manager		
Danish	22	22
Finnish	0	9
Finnish	3	10
Hungarian	0	6

Table 3c. Customer background

Customer		
Italian	0	2

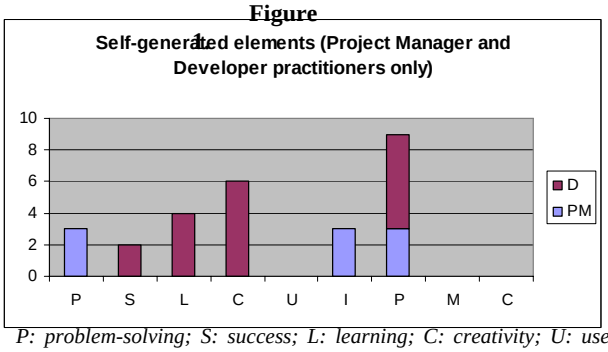
Table 3d. Mentor/coach background

Mentor/coach		
Finnish	6	20+
UK	30	4

4.2 Elements generated within role groups

Figure 1 shows the motivation elements generated by developers and project managers. Each element has been classified into one of the 9 classifications as shown.

Figure 1



P: problem-solving; S: success; L: learning; C: creativity; U: users; I: influence; P: people; M: making a difference; C: clear presentation

Figure 1 suggests variations in motivation elements that developers and project managers generated. Figure 1 shows that elements from project managers only fell into three categories: *problem-solving*, *influence* and *people*, with the first two being only generated by project managers. Elements from developers fell into four categories: *success*, *learning*, *creativity* and *people*, with the first three being generated only by developers. *People* is the theme with most elements in it and is the only theme contributed to by both developers and project managers.

4.3 Constructs generated within role groups

The findings presented in this section relate to the constructs developers and project managers identified for the 3 pre-set element triads shown in Table 1.

4.3.1.1 Triad 1: technical challenge; good tools; rewards and benefits

Figure 2. Triad 1 Developers and project manager constructs

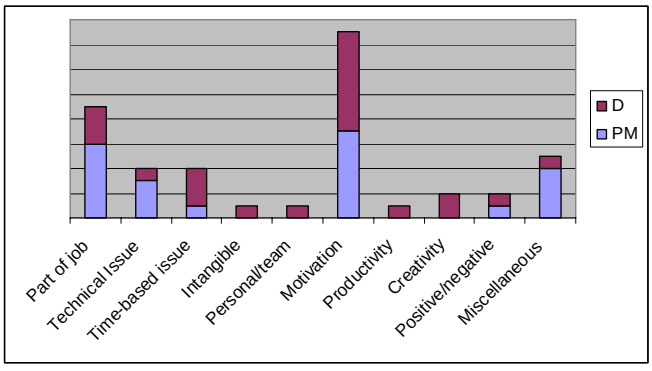
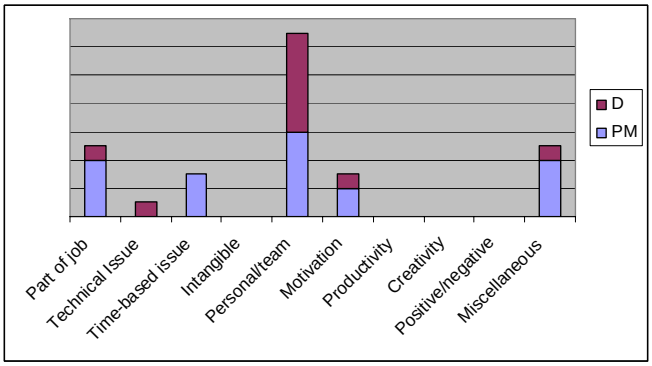


Figure 2 shows the themes that emerged in the bi-polar constructs developers and project managers identified in relation to the elements in Triad 1. Again Figure 2 suggests distinctions in the way developers and project managers think about motivation. In particular Figure 2 shows that only developers identified constructs concerned with *creativity*, *productivity*, *intangible* and *personal/team*. Figure 2 also shows a very different profile of constructs for project managers who focused their constructs on Triad 1 elements being *part of the job* and *technical issues*. Overall the largest number of constructs for this triad related to the theme of *motivation*. Furthermore the number of constructs related to motivation for both developers and project mangers is significantly higher for this triad of elements than for the other 2 triads. This suggests that these elements may be particularly important elements for motivation.

4.3.1.2 Triad 2: working with people; recognition; ownership

Figure 3 shows the classification of bi-polar constructs for Triad 2. Figure 3 shows that the largest construct theme is *personal/team* and is the only theme where developers and project managers contribute nearly equally. Figure 3, again, suggests differences between developers and project managers. Project managers perceived elements in this triad as more *part of the job* and more *time-based.* A small number of developer constructs were *technical issues* with no project manager constructs similarly classified.

Figure 3. Triad 2 Developers and project manager constructs



4.3.1.3 Triad 3: autonomy; job security; career prospects

Figure 4. Triad 3 Developers and project manager constructs

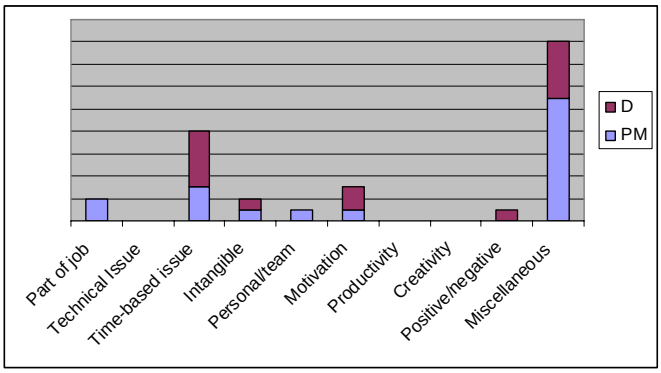
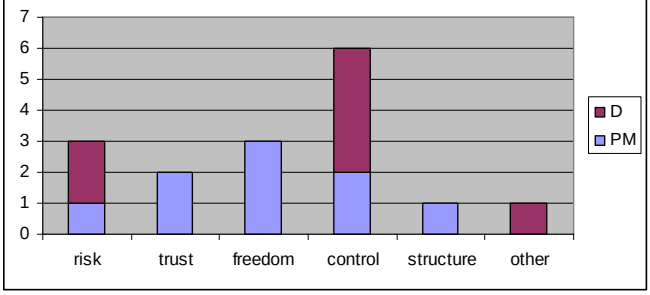


Figure 4 shows less variation between developers and project managers in relation to their perceptions of these elements. Figure 4 also shows that, apart from *miscellaneous*, the *time-based* theme contains the largest number of constructs for these elements with developers and project managers contributing nearly equally to this theme. Triad 3 has the lowest number of constructs in the *motivation* theme of all the triads and suggests that the elements in this triad may be less central to motivation than those in the

other 2 triads. Figure 4 shows that triad 3 contains may constructs classified as *miscellaneous.* We have analysed these constructs at a finer level of granularity as shown in Figure 5.

Further analysis of the miscellaneous classification again suggests variation between developers and project managers. Figure 5 shows that only project managers cite constructs related to *trust, freedom* and *structure.* Figure 5 also shows similarity between developers and project managers on *risk* and *control* in relation to the elements in triad 3.

Figure 5. Triad 3 analysis of *miscellaneous* theme



4.4 Grid analysis for Triads 1, 2 and 3

We now present a sub-set of the grid analysis for each triad in relation to the following 3 aspects of analysis:

A. Where does agile rate?
B. Where does plan-driven rate?
C. Where do you rate?

It is impractical to present the full grid for every construct that every individual did. This is because unlike the element and construct data presented above, it is not possible to compress grids by classifying data into themes. In grids every single construct within every single triad has a score associated with it. Compressing this data loses much of the power of the grid for an individual. We therefore present grid excerpts for raw constructs within two prominent themes, together with their alignment scores. Grid excerpts have been compressed for presentation. An uncompressed example grid is shown below:

'*Where does your current project rate on the grid*?'

Bi-polar extreme	Alignment							Bi-polar extreme
Employed by a company	1 ✓	2	3	4	5	6	7	Doing voluntary work
Developing commercial software	1	2	3 ✓	4	5	6	7	Developing free software
Young	1	2	3	4	5	6 ✓	7	Old

For brevity we have compressed the presentation of such full grids into the grid shown below:

Bi-polar extreme	Grid score	Bi-polar extreme
Employed by a company	1	Doing voluntary work
Developing commercial software	3	Developing free software
Young	6	Old

4.4.1.1 Triad 1: grid analysis – motivation theme

Tables 4a-c show all bi-polar constructs developers and project managers generated for triad 1 (T1) that were classified under the *motivation* theme. Each separate table analyses one of the 2-to-1 element configurations. For example Table 4a shows a grid excerpt where elements are considered in the configuration: what is the same about *technical challenge (E1)* and *good tools (E3)* that makes them different from *rewards and benefits (E1)*? Whereas Table 4b considers the configuration of elements: What is the same about E1 and E3 that makes them different from E2.

Table 4a. Grid for T1 E2&E3vE1

← Raw bi-polar constructs →	A	B	C	
	Grid Score 1……7			
Not connected to excitement	4	1	7	Exciting **D**
boring	7	2	7	Interesting **D**
boring	6	2	4	Interesting **D**
helping	-	-	-	Exciting **D**
cannot be too much	2	7	5	when it's too much it's **PM** demotivating

A. Where does agile rate? B. Where does plan-driven rate? C. Where do you rate?

The grid scores shown in Table 4a suggest that developers and the project manager rate agile approaches as more motivating (in terms of their constructs related to interest and excitement) than plan-driven approaches. There are consistently large differences in the grid scores given to each approach. Table 4a also suggests that there is no consistency in how practitioners align their own motivation constructs. Two practitioners give themselves scores of 7 while the others score themselves 4 and 5.

Table 4b. Grid for T1 E1&E3vE2

← Raw bi-polar constructs →	A	B	C	
	Grid Score 1……7			
motivating	1	4	-	not motivating **D**
motivating	-	-	-	helping work **D**
cause anxiety	6	4	6	relieve anxiety **D**
cause anxiety	3	1	2	relieve anxiety **D**
makes you feel good	1	6	1	somebody gives **PM**
stimulating	2	6	2	Supporting **PM**
motivators	2	7	6	Effectiveness **PM**

A. Where does agile rate? B. Where does plan-driven rate? C. Where do you rate?

Table 4b also suggests differences between perceptions of agile and planned approaches in terms of general motivation constructs (ie constructs related to motivation, relieving anxiety, stimulation and feeling good). Agile is fairly consistently rated closer to the positive constructs on the grid than plan-driven. Table 4b does not show any trends in developers or project managers aligning themselves particularly with either approach.

Table 4c suggests some difference between how developers perceive agile and plan-driven in relation to job satisfaction constructs. However our results are not as strong as the literature

would suggest. Developers are also generally aligning themselves in the middle of the two approaches in terms of job satisfaction constructs (ie constructs related to satisfaction, staying in the job and ease of work). Table 4c suggests that project managers in this data set perceive agile and plan-driven as very different. A particularly interesting project manager perception is that agile is further away from effectiveness than plan-driven. Table 4c also suggests that for these constructs, project managers are aligning themselves in the middle of agile and plan-driven.

Table 4c. Grid for T1 E1&E2vE3

← Raw bi-polar constructs →				
	A	B	C	
	Grid Score 1…….7			
work satisfaction	2	5	3	reason to work **D**
makes me stay on the job	2	6	4	makes me stay in the job **D**
job satisfaction	2	4	4	money, reason for work **D**
easier to work for	7	2	5	Result **PM**
effectiveness	6	1	4	Satisfaction **PM**
stay in project	-	-	-	stay in company **D**

A. Where does agile rate? B. Where does plan-driven rate? C. Where do you rate?

4.4.1.2 Triad 2: grid analysis – personal/team

Tables 5a-c show all bi-polar constructs developers and project managers generated for triad 2 that were classified under the *personal/team* theme.

Table 5a. Grid for T2 E2&E3vE1

← Raw bi-polar constructs →				
	A	B	C	
	Grid Score 1…….7			
star	7	5	2	Team **D**
lone stars	5	2	5	group work **D**
management associated	4	1	4	colleagues related **PM**

A. Where does agile rate? B. Where does plan-driven rate? C. Where do you rate?

Table 5a suggests consistent differences between agile and plan-driven in terms of constructs in the *personal/team* theme. Agile is perceived to be more team oriented while plan more associated with 'stars'. There is no particular pattern in this data set showing how people rate themselves in this regard.

Table 5b. Grid for T2 E1&E3vE2

← Raw bi-polar constructs →				
	A	B	C	
	Grid Score 1…….7			
connected	2	5	1	Independent **PM**
team work	3	3	3	Personal **D**
more for team	4	4	6	more personal **D**
enables	3	7	1	reliance of
independence				outsiders **D**
enables independence	3	5	3	accepts **D** independence

A. Where does agile rate? B. Where does plan-driven rate? C. Where do you rate?

Table 5b suggests less variation in how developers perceive agile and plan-driven in terms of the constructs presented. Although there is a very slight trend to developers aligning themselves to agile, this is not a strong trend.

Table 5c. Grid for T2 E1&E2vE3

← Raw bi-polar constructs →				
	A	B	C	
	Grid Score 1…….7			
team work	1	3	7	Lonely **D**
communication & interaction	-	-	-	being alone **D**
team work	2	6	2	working alone **D**
interaction	2	4	4	being alone **D**
group work	2	5	3	Individual **PM**
group work	2	5	2	Individual **PM**
individual	7	5	1	can be shared **PM**
dependence	2	2	5	Independence **PM**

A. Where does agile rate? B. Where does plan-driven rate? C. Where do you rate?

Table 5c suggests some variation between perceptions of agile and plan-driven. Table 5c does not show any particular trends regarding participants aligning themselves to a particular approach. Furthermore Table 5c does not show particularly different perceptions between project managers and developers.

4.5 Threats to Validity

There are a number of threats to the validity of the findings reported here. The most significant is the small amount of data we report. As a consequence our findings cannot be generalized. They simply offer interesting insights of the perceptions that a small number of developers and project managers had at the XP2006 workshop. We are currently collecting more data using our RGT approach which we will use to extend our data set and validate the findings we report here.

A further threat is related to the self-selecting nature of the sample we report. Clearly our sample is biased to practitioners who not only attended XP2006, but also chose to participate in a workshop on motivation. Again the findings we report here must be extended to a more representative sample of practitioners. We are about to repeat this RGT exercise at a non-agile conference (SPA2007). This will enable us to identify agile bias in our data set and compare the attitudes of agile to plan-driven practitioners.

There are methodological issues related to our use of RGT that may compromise the purity of the findings we report here. In particular the workshop setting in which we ran the RGT exercise meant that participants were exposed to ideas of motivation

before and during the exercise. There is a chance that these ancillary discussions may have influenced participants in their thinking.

5. DISCUSSION

Although this study presents data collected from a very small sample of developers and project managers, several promising strands of further enquiry have emerged from our analysis.

5.1 The importance of some motivators

We did not design this study to identify the important aspects of motivation. We simply used as elements in our RGT exercise motivators identified in the literature as relevant to software practitioners. Indeed the literature currently does not identify a ranked list of motivators for software practitioners. However our findings do suggest that some aspects of motivation are more important to software practitioners than other aspects. We make this judgment on the basis that some elements generated far more constructs directly related to motivation than other elements. This is particularly the case with the elements in triad 1: technical challenge; good tools; rewards and benefits. This is interesting as this triad contains elements intrinsically related to the job of software engineering, as well as an extrinsic factor related to the job. Elements in the other 2 triads are more related to general motivating features of any job. This suggests that software practitioners actually find software engineering tasks motivating. We selected all elements initially from the literature, so it is not surprising that the importance *of technical challenge* and *good tools* has been previously reported (Tanner, 2003; Ramachandran and Rao, 2006). However what is not reported currently in the literature is that these motivators may be more important to software practitioners than other motivators (for example, autonomy). Furthermore the apparent importance of *rewards and benefits* is not cited in the literature as highly significant to software practitioners. Our findings may be a peculiarity of our small biased data set, but on the other hand, may suggest that how software practitioners are motivated is shifting over time.

Our findings on the importance of some motivators are also relevant to the main focus of this study. If intrinsic aspects of software engineering together with rewards and benefits are more relevant to motivating developers than the more general aspects of the job, then it is vitally important that managers understand this. Such understanding can then be used to improve the way developers are managed.

5.2 Motivators for developers and project managers

Our results suggest that developers and project managers seem to perceive motivators differently. Project managers identified a generally different set of motivational elements from developers. Project managers also generated a generally different profile of constructs to developers in response to the pre-set triads of motivational elements. Less difference was observable in the grids project managers and developers constructed, though this may reflect the small data set.

This finding is not entirely unexpected. Generic theories of motivation suggest that different motivators are relevant to people at different stages in their career. For example Maslow's hierarchy of needs theory (Maslow, 1954) says that people will have different needs at different points in their life. This means they will find different things motivating at different times. It is likely that developers are at different points in their life to project managers, and so Maslow's hierarchy of needs may explain why they are motivated by different things.

The most important implication of our findings on the differences between developers and project managers in terms of their motivations, is that project managers should recognize these differences. It may be that project managers assume that developers are motivated in a similar way to themselves. Our findings suggest that such a management approach would be ineffective. Indeed our findings suggest that there is more work needed to identify profiles of motivators that a variety of role groups within software engineering may have. Work in this area was last done many years ago (eg. Goldstein and Rockart (1984)). Understanding role specific motivations could underpin a multi-dimensional approach to getting the best work out of people. Furthermore such an approach to designing management strategies would be easier to implement than trying to tailor management strategies around, for example, individual personality difference - which is the most common conclusion from research in the area.

Our previous research on software process improvement also shows that developers and project managers often have very different responses to organizational initiatives. Our previous work has recognized that multi-stranded management strategies are needed to match the needs of different role groups.

5.3 Agile and plan-driven motivators

Our grid analysis identified some interesting points of further research in terms of how agile and plan-driven approaches relate to motivation. Although the data in this grid is too sparse to be significant, it suggests that developers and project managers both perceive agile as more aligned to motivation and job satisfaction than plan-driven. This finding confirms what the literature says about agile approaches (as discussed above). However the agile experience of our participants is not tremendously high so it's difficult to know the basis on which their opinion has been formed.

Our data also suggests that developers align themselves more than project managers to agile. This apparent mis-alignment between developers and project managers highlights important areas of further research and could have important implications for the effective running of software projects, particularly if these mis-alignments go unrecognized and un-resolved.

6. CONCLUSIONS

Previous studies have found that highly motivated software practitioners make a difference to project success. Consequently it is very important that project managers are able to manage projects to maximize developers' motivation. Our results suggest that the motivators of project managers and developers are probably different. It is, therefore, vitally important that project managers are aware that what motivates themselves does not necessarily motivate developers. It is equally important that project managers are aware of the motivational profile of developers. Our results suggest that this profile may be changing and that the most important motivators are related to the job of software engineering in terms of the technical challenge and the

tools. However similarly important are rewards and incentives. Our results do suggest that these motivators are equally valued by developers and project managers.

7. ACKNOWLEDGMENTS

This work was supported by the UK's Engineering & Physical Sciences Research Council under grant number EP/D057272/1.

We would like to thank the other co-chairs of the XP2006 workshop at which this data was collected: Bjørnar Tessem, Frank Maurer, Daniel Karlström, Yvonne Dittrich. We would also like to thank all the participants in the workshop.

8. REFERENCES

Baddoo, N. and Hall, T. (2003). De-motivators for software process improvement: An analysis of practitioners' views, *Journal of Systems and Software* 66 (1): 23-33. Elsevier Inc.

Beecham, S., Baddoo, N., et al. (2007). Motivation in Software Engineering: A Systematic Literature Review (in review), *ACM Computing Surveys*.

Beecham S, Hall T, et al (2003) Software Process Improvement Problems in 12 Software Companies: An Empirical Analysis, *Empirical Software Engineering* 8(1): 7-42.

Beecham, S., Sharp H, Baddoo, N., Hall T, Robinson H (2007). Does the XP environment meet the motivational needs of the Software Developer? An Empirical Study, Agile Conference, in review

Boehm, B.W., *Software Engineering Economics*. advances in computing science and technology series, ed. Prentice-Hall. 1981, Englewood Cliffs: Prentice-Hall, Inc.

Capretz, L., *Personality Types in Software Engineering.* International Journal of Human-Computer Studies, 2003. 58(2): p. 207-214.

DeMarco, T. and T. Lister, *Peopleware - Productive Projects And Teams.* 1999.

Downey, J. (2006). Systems architect and systems analyst: are these comparable roles? *Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research*

Enns, H. G., Ferratt, T. W., et al. (2006). Beyond Stereotypes of IT Professionals: Implications for IT HR Practices, *COMMUNICATIONS OF THE ACM* 49 (4): 106-109.

Fransella, F. and D.e. Bannister, *A Manual For Repertory Grid Technique*. 1977: London Academic Press.

Goldstein, D. K. and Rockart, J. F. (1984). An Examination of Work-related Correlates of Job Satisfaction in Programmer/Analysts, *MIS Quarterly* 8 (2): 103-115.

Hackman, J. R., Oldham, G. R., et al. (1975). A new strategy for job enrichment, *California Management Review* 17: 57-71.

Jekielek, J. (2002). Control Systems Professional: A Transition from the Technical to the Managerial Role, *ISA TECH/EXPO Technology Update Conference Proceedings* 424-425: 631-639. ISA - Instrumentation, Systems, and Automation Society.

Kaarst-Brown, M. L. and Guzman, I. R. (2005). Who is "the IT workforce"?: challenges facing policy makers, educators, management, and research, *Proceedings of the 2005 ACM SIGMIS CPR Conference on Computer Personnel Research (Atlanta, Georgia, USA, April 14 - 16, 2005). SIGMIS CPR '05*: 1-8. ACM Press.

Mannaro, K., M. Melis and M. Marchesi, Empirical Analysis on the Satisfaction of IT Employees Comparing XP Practices with Other Software Development Methodologies. Lecture Notes in Computer Science, 2004: p. 166-174

McConnell, S. (1998). Problem programmers, *Software, IEEE* 15 (2): 128, 127, 126.

McConnell, S., *Avoiding classic mistakes [software engineering]*. IEEE Software, 1996. 13(5): p. 111-112.

Melnik, G. and F. Maurer, *Comparative Analysis of Job Satisfaction in Agile and Non-Agile Software Development Teams.* XP2006, 2006

Myers, M. E. (1991). Motivation and performance in the information systems field: a survey of related studies *SIGCPR Comput. Pers.* 13 (3): 44-49. ACM Press.

Orlikowski, W. and Baroudi, J. J. (1989). The Information Systems Profession: Myth or Reality?, *Office: Technology and People* 4: 13-30.

Prasad, J., Ferratt, T. W., et al. (2005). One Size Does Not Fit All: Managing IT Employees' Employment Arrangements, *http://www.sba.udayton.edu/research/working_papers/WP17.pdf*. School of Business Administration

Procaccino, J.D., J.M. Verner, K.M. Shelfer and D. Gefen, *What do software practitioners really think about project success: An exploratory study.* Journal of Systems and Software, 2005. 78(2): p. 194-203.

ProjectLink (2006) *Motivation House*. Available from: http://www.projectlink.co.uk/whoweworkfor.htm, accessed 12.5.2006.

Ramachandran, S. and S.V. Rao, *An effort towards identifying occupational culture among information systems professionals.* Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research: Forty four years of computer personnel research: achievements, challenges & the future. Claremont, California, USA, 2006: p. 198-204.

Robson, C. (2002) Real World Research (2nd edition), Blackwell Publishing.

Sharp H, Hall T, Tessem B, Maurer F, Karlström D, Dittrich Y, *Human & Social Factors in Software Engineering: motivation and de-motivation in agile development,* XP2006, Oulu, Finland, 2006, June

Shaw, M.L.G. and B.R. Gaines, *Requirements acquisition.* Software Engineering Journal, 1996. **11**(3): p. 149-165.

Standish Report (1995) *Standish Group Chaos Report.* URL: http://www.scs.carleton.ca/~beau/PM/Standish-Report.html

Syed-Abdullah, S.L., J. Karn, M. Holcombe, T. Cowling and M. Gheorge, *The positive affect of the XP methodology*. Extreme Programming and Agile Processes in Software Engineering. 6th International Conference, XP 2005. Proceedings (Lecture Notes in Computer Science Vol. 3556). Springer-Verlag. 2005, 2005: p. 218-21.

Tanner, F.R., *On motivating engineers.* Engineering Management Conference, 2003. IEMC '03. Managing Technologically Driven Organizations: The Human Side of Innovation and Change, 2003: p. 214-218.

Thomas, L. and E.S. Harri-Augstein, *Self-Organized Learning: Foundations of a Conversational Science for Psychology*. 1985, London: Routledge Keegan Paul.

Whitaker, K. (1997). Motivating and keeping software developers, *Computer* 30 (1): 126-128.