# Exploring Motivational Differences between Software Developers and Project Managers

Helen Sharp

Maths and Computing Faculty
Open University
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of software engineers is generally poorly understood and previous work done in the area is thought to be largely out-of-date.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.

## 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 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Some studies suggest that conventional approaches to motivation in software engineering might be outdated.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). 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 (Tanner 2003; Procaccino et al. 2005). We found previously (Beecham et al, 2007) an increasing awareness of the importance of motivating software developers.

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 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.

## 2. RESEARCH METHODS
### 2.1 Repertory Grid Technique

We use the repertory grid technique (RGT) (Fransella and Bannister 1977) to collect and analyse software practitioners' perception of motivation. The aim of RGT is to allow participants to reveal their beliefs without any influence from the researcher. 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). 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.

**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.

**Bi-polar constructs**. RGT aims to elicit from participants constructs that distinguish between elements. 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.

## 2.2 Our Implementation of RGT

### 2.2.1 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.

### 2.2.2 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.

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

### 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 |

### 2.2.4 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:

| Element categories | | |
| --- | --- | --- |
| Problem-solving | Creativity | People |
| Success | Users | Making a difference |
| Learning | Influence | 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.

All the individual constructs that participants generated were grouped together. Themes (listed under figures 2-4) 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 |

## 3. FINDINGS

### 3.1 Participants
In this study we are particularly interested in practitioner motivation. Consequently we focus on data from 4 developers and 6 project managers. All data we present for developers and project managers has been normalized to account for the imbalance in the representation of each of the two groups.

### 3.2 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.
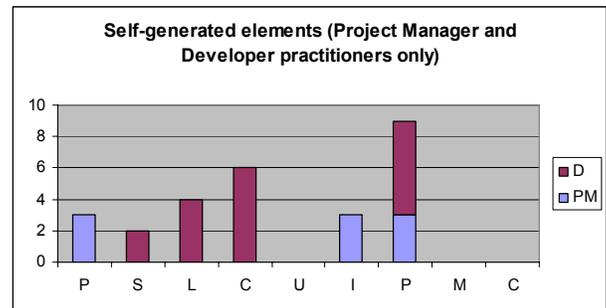


**Figure 1.**

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.

### 3.3 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.

## 3.3.1.1 Triad 1: technical challenge; good tools; rewards and benefits
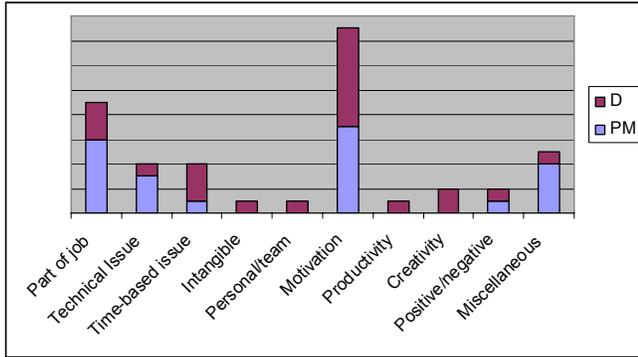


**Figure 2. Triad 1 Developers and project manager constructs**

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; they 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.

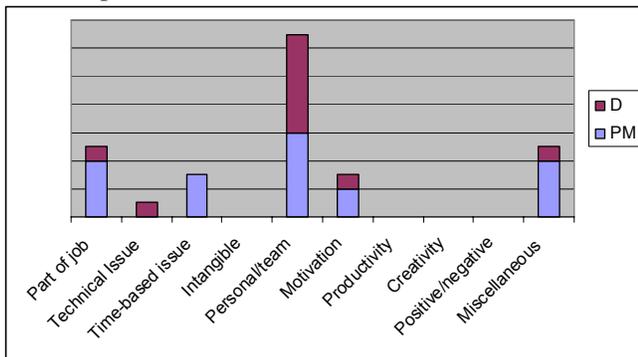## 3.3.1.2 Triad 2: working with people; recognition; ownership



**Figure 3. Triad 2 Developers and project manager constructs**

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.*

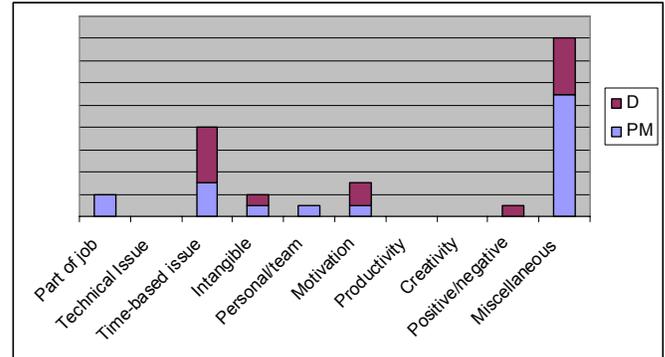## 3.3.1.3 Triad 3: autonomy; job security; career prospects



**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.* A more detailed analysis of this category shows that only project managers cite constructs related to *trust, freedom* and *structure* while developers and project managers both identify constructs related to *risk* and *control.*

## 4. 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.

## 4.1 The importance of some motivators

Our findings 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.

## 4.2 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.

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.

## 5. 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. REFERENCES

[1] 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, Washington, IEEE CS Press

[2] Boehm, B.W., *Software Engineering Economics*. Prentice-Hall. 1981, Englewood Cliffs: Prentice-Hall, Inc.

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

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

[5] 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.

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

[7] 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.

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

[9] Ramachandran, S., S.V. Rao, *An effort towards identifying occupational culture among information systems professionals*. ACM SIGMIS CPR conf on computer personnel research: Claremont, 2006: p. 198-204.

[10] 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

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

[12] 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. XP2005. Procs (Lecture Notes in Computer Science Vol. 3556). Springer-Verlag. 2005, 2005: p. 218-21.

[13] Tanner, F.R., *On motivating engineers*. Engineering Management Conference, 2003. IEMC  p. 214-21