MOTIVATING SOFTWARE ENGINEERS
A Theoretically Reflective Model

Nathan Baddoo, Sarah Beecham, Tracy Hall
School of Computer Science, University of Hertfordshire, College Lane, Hatfield, Herts AL10 9AB, U.K.

Hugh Robinson, Helen Sharp
Dept of Computing, Faculty of Mathematics and Computing, The Open University, MK7 6AA, U.K.

Keywords: Motivation, model, theory, software engineer, systematic literature review.

Abstract: We present a model of motivation for software engineers. Our model suggests that software engineers are motivated by two sets of factors, intrinsic and extrinsic motivators, where a subset of intrinsic motivators are aspects inherent in the job that software engineers do. It shows that software engineers are orientated towards these particular sets of motivators because of their characteristics, which in turn are mediated by individual personality traits and environmental factors. Our model shows that the external outcomes of software engineers' motivation are benefits like staff retention, increased productivity and reduced absenteeism. Our model is derived from a Systematic Literature Review of motivation in software engineering. We have constructed this model by engaging in practices that reflect good principles of model building as prescribed by operational research and scientific management discipline. We evaluate our model for theoretical efficacy and show that our model, in comparison to other attempts at modeling software engineers' motivation, reflects a wide range of the classic concepts that underpin the subject area of motivation. We argue that, this theoretical efficacy validates the model and therefore improves confidence in its use. We suggest that our model serves as a valuable starting point for managers wanting to understand how to get the best out of software engineers, and individuals wanting to understand their own motivation or who are embarking on career choice.

1 INTRODUCTION

In this paper we present a model of motivation for software engineers. We derive this model from a Systematic Literature Review (SLR) of software engineers' motivators. We follow model-building principles from the discipline of operational research and scientific management to construct this model and evaluate the model in terms of how it reflects classic theories of motivation. We show that our model is strongly corroborated by existing phenomena in the organizational psychology literature on motivation.

Our model of motivation is based on findings from a systematic literature review of 92 published papers. This means it is rigorously underpinned by previous work in the area. Our model shows that there is a complex array of factors that must be managed effectively to get the best outcomes from software engineers. Our model also reflects some classic concepts of motivation, which is an important outcome of this work, since our related work suggests that theories are not used well in existing studies of motivation of software engineers (Hall et al 2007).

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 (DeMarco & Lister 1999). The Standish report (1994) amplifies this finding by reporting that having access to competent, hard working and focused staff is one of 10 success criteria for software projects. However, until now no comprehensive model of what motivates software engineers has been developed. Consequently it is difficult for managers to know how best to motivate their software engineers.

Baddoo N., Beecham S., Hall T., Robinson H. and Sharp H. (2008).
MOTIVATING SOFTWARE ENGINEERS - A Theoretically Reflective Model.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - HCI*, pages 174-180
DOI: 10.5220/0001685601740180

In Section 2, we provide some background on the value of models. We also present an overview of motivation theories. In Section 3, we describe our research method. In Section 4, we present the findings of our SLR, detailing how these findings underpin the development of our model. In Section 5, we present our model and evaluate it in terms of how it reflects classic and conventional concepts of motivation in. We conclude and discuss future work in Section 6.

2 THE VALUE OF MODELS

Any model derived to examine a particular area of study or practice should reflect the classic theories in that area. Indeed, work that we have done shows that some models are actually representations of theory. We provide a summary of theories of motivation. An extensive overview of these theories is published by Hall et al (2007). We will, in later sections, reflect the model presented in this paper in the light of these theories.

2.1 Why are Models Important?

Models are used in all fields of software engineering, from requirements engineering to software evolution either to explore possible consequences of an action before taking that action or as embedded parts of a system to aid in routine decision making (Pidd 1999). Overall, our exploration for a model to represent software engineers' motivation aims at deriving some form of external representation of the dynamics of how software engineers are/can be motivated. Such a model will be important to the people who manage software engineers in terms of helping them to manager software engineers better.

2.2 Concepts of Motivation

A good model of motivation should address some classic theories of motivation. There are eight theories that specifically address how people are motivated within an organizational context. We provide an overview of these as follows

2.2.1 Job Characteristics Theory

The Job Characteristics Theory (JCT) states that there are certain key characteristics present within a job that makes it motivational to practitioners (Hackman and Oldham, 1976). These key characteristics are classified into five 'core dimensions', which are Skill variety, Task identity, Task significance, Autonomy and Feedback. The extent to which these five job dimensions motivate practitioners is dependent on his need for personal growth and development; Growth Need Strengths (GNS) (Hackman and Oldham, 1976).

2.2.2 Stimulus Response Theory

Stimulus Response Theory (SRT) describes the activities that modify behaviour (Skinner, 1976). These activities are termed stimuli. According to SRT there are two types of stimuli: punitive and rewarding stimuli. The theory explains that punitive stimuli are easier to apply and do have the effect of producing the required responses in the short term. However, rewarding stimuli, which are more difficult to apply and require more ingenuity to devise, tend to have a longer-term effect in inducing the correct responses from subjects (Skinner, 1976).

2.2.3 Equity Theory

Adam's Equity Theory (1963), in an organization context, is concerned with how to make employees feel equitably treated in an organisation. It states that the inputs that people bring into an activity or organisation, that is their experience, education, skills and seniority, should be matched by the outputs that is what they get from that activity or organisation, which are salary, recognition, opportunity for achievement etc.

2.2.4 Needs Theory - Maslow

Maslow's hierarchy of needs roughly translates that different types of needs motivate people at different stages in their lives (Maslow, 1954). Such needs manifest in a hierarchy where physical needs are at the bottom of the hierarchy and self-actualisation comes at the top.

2.2.5 Need Theory McClleland

McClelland's needs theory identifies three motivational needs: achievement, authority and affiliation (McClelland 1961). It states that each individual has a combination of these needs in various levels of strength. So that the individual whose need mix is strongly biased by affiliation will tend to be a more objective in order to increase their opportunity for bonding with the most number of people.

The interesting distinction between McClelland's theory and Maslow's is the absence of the hierarchical structure on which these needs manifest. However, it is possible to assume that with time, an individual's needs mix will change and assume different profile depending on where he finds himself.

2.2.6 Motivation-Hygiene Theory

Herzberg's motivation hygiene theory classifies factors that motivate practitioners into two distinct sets: Extrinsic factors and intrinsic factors.

- Extrinsic factors are those that are external to the job that practitioners do.
- Intrinsic factors, on the other hand, are the primary determinants of motivation and satisfaction.

2.2.7 Goal Setting Theory

This theory states that goals that are hard to achieve, when accepted, lead to better performance by the people doing them than goals that are easy to achieve (Locke 1968). However, in order to do this, the goal needs to be very well defined, made specific and measurable, and feedback provided so that the person tackling the goal will know when it is achieved.

2.2.8 Expectancy Theory

Much of Vroom's (1964) expectancy theory is based around the notion that a individual's motivation to engage in certain activity is predicated by the degree or amount of positive outcomes that he expects from this activity.

3 RESEARCH METHOD

We analysed the research problem in relation to two main question areas:

- How are software engineers motivated?
- What models of motivation exist in software engineering to explain how software engineers are motivated?

We conducted a SLR, following the guidelines of Kitchenham (2004). We formulated the following set of research questions:

RQ1: What are the characteristics of Software Engineers?

RQ2: What (de)motivates Software Engineers to be more (less) productive?

RQ3: What are the external signs or outcomes of (de)motivated software Engineers?

RQ4: What aspects of Software Engineering (de)motivate Software Engineers?

RQ5: What models of motivation exist in Software engineering

The rest of the procedure followed in the SLR can be found in (Beecham et al 2006).

3.1 Overview of Model Construction Process

Pidd et al suggest that a good model must be simple. Such simplicity can be derived from how transparent the model turns out to be. Transparency, in turn, can be achieved by how well defined the problem is. In our model construction the problem we explore is tightly defined by the research questions formulated for the SLR. RQ1 to RQ4 provide us with a clear definition of the building blocks to the overall question on what motivates software engineers. The aim is that the solutions to the questions would provide the rationale for and the parameters for building the model. The overall structure of the model itself will be derived from similarities of other tried and tested work.

The rationale for building our model, in light of the research that indicates that models are important in helping us think and reflect about a phenomenon before acting, is to establish whether such a model already exists. In the context of this research, this line of enquiry is catered for by the inclusion of Research Question 5, above, in the SLR. The answers to RQ5 should help us understand what the gaps are in the area of models of software engineers' motivation.

The next stage is to simplify the problem by decomposing it into simple components. This process involves a lot of knowledge of the problem area. This requires that in formulating questions that will provide solution to the line of enquiry, we also needed to know how that problem area could be decomposed into simple modules. In doing this, we are also ensuring that should certain components of the resultant enquiry be found to be erroneous, they can be replaced without having to replace or re-do the entire model.

This simplification through decomposition is provided by the breakdown of the research enquiry into software engineers' motivation into four research question. Figure 1 illustrates this point.

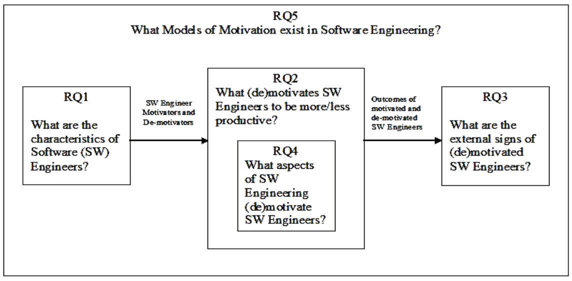
RQ5
What Models of Motivation exist in Software Engineering?
RQ1
What are the characteristics of Software (SW) Engineers?
SW Engineer Motivators and De-motivators
RQ2
What (de)motivates SW Engineers to be more/less productive?
RQ4
What aspects of SW Engineering (de)motivate SW Engineers?
Outcomes of motivated and de-motivated SW Engineers
RQ3
What are the external signs of (de)motivated SW Engineers?

Figure 1: Framework of research strategy for SLR (Beecham et al 2007).

The resultant data from the chain of enquiry represented in Figure 1 allowed us to parameterise the components of the individual modules of the emerging model and the overall structure of the model was then derived from analogies and similarities with other tried and tested work on generic concepts of motivation and by analysing other models of motivation in software engineering. We explain the tried and tested work and other models of motivation in Section 5, when we present and discuss the model.

The final stage in the model construction exercise was to ascertain how the resultant model reflected both classic and conventional theories of motivation.

4 WHAT MOTIVATES SOFTWARE ENGINEERS?

In the following section, we present findings from the systematic literature review to the Research Questions 1 to 5 presented earlier.

4.1 What are the Characteristics of Software Engineers?

Table 1 shows that the SLR reports a software engineer to be *growth oriented,* i.e. he is an individual who likes challenges and likes to learn new skills. It also shows that the SLR reports a software engineer to bean *introverted* individual with low need for social strengths. The SLR also reports a software engineer to be *autonomous*, *creative* and *technically competent*.

Table 1: Software Engineer Characteristics (Beecham et al 2007).

Software Engineer Characteristics	%
Ch.4 Growth oriented (e.g. challenge, learn new skills)	9
Ch.6 Introverted (low need for social interaction)	8
Ch.11 Autonomous (need for independence)	8
Ch.1 Need for stability (organisational stability)	5
Ch.3 Achievement oriented (e.g. seeks promotion)	4
Ch.12 Need for variety	4
Ch.14 Need for challenge	4
Ch.16 Need to be sociable/identify with group/organisation	4
Ch.2 Technically competent	3
Ch.5 Need for competent supervising	3
Ch.8 Need for feedback (needs recognition)	2
Ch.10 Need to make a contribution (job is worthwhile)	2
Ch.13 Marketable	2
Ch.15 Creative	2
Ch.7 Need for involvement in personal goal setting	1
Ch.9 Need for Geographic stability	1

4.2 What (de)Motivates Software Engineers to be more (less) Productive?

Table 2 shows that some of the most widely reported motivators of software engineers from the SLR are the ability to *identify with the tasks*, *employee participation*, *career paths*, *good management*, *variety of work* and a *sense of belonging*.

Table 2: What motivates Software Engineers? (Beecham et al 2007).

Motivators in Software Engineering	%
M.17 Identify with the task (clear goals, personal interest, know purpose of task, how it fits in with whole, job satisfaction; identifiable piece of quality work)	2
M.10 Employee participation/involvement/working with others	17
M.4 Career Path (opportunity for advancement, promotion prospect, career planning)	16
M.6 Good management (senior management support, *team-building, good communication)	16
M.3 Variety of Work (e.g. making good use of skills, being stretched)	15
M.7 Sense of belonging/supportive relationships	15
M.1 Rewards and incentives (e.g. scope for increased pay and benefits linked to performance)	14
M.12 Recognition (for a high quality, good job done -different to M1 which is about making sure that there are rewards available).	12
M.2 Development needs addressed (e.g. training opportunities to widen skills)	11

Table 2: What motivates Software Engineers? (Beecham et al 2007)(cont).

Motivators in Software Engineering	%
M.11 Feedback	11
M.15 Technically challenging work	11
M.16 Job security/stable environment	11
M.18 Autonomy	9
M.8 Work/life balance (flexibility in work times, caring manager/employer, work location)	7
M.21 Making a contribution/task significance (degree to which the job has a substantial impact on the lives or work of other people)	7
M.5 Empowerment/responsibility	5
M.19 Appropriate working conditions/environment/good equipment/tools	5
M.14 Trust/respect	4
M.13 Equity	3
M.9 Working in company that is successful (e.g. financially stable)	2
M.22 Sufficient resources	2

4.3 What are the External Signs or Outcomes of (de)Motivated Software Engineers?

Table 3 shows that the most widely reported outcome of motivated software engineers is *retention*. Other external outcomes are improvements in *project delivery time* and *productivity*, adherence to *budgets*, low *absenteeism* and improved *project success*.

Table 3: External signs of (de)motivated software engineers (Beecham et al 2007).

External signs of motivated and de-motivated software engineers	% of studies
Ext1: Retention	11%
Ext2: Project delivery time	2%
Ext3: Productivity	5%
Ext4: Budgets	1%
Ext5: Absenteeism	1%
Ext6: Project Success	1%

4.4 What are Aspects of Software Engineering (de)Motivate Software Engineers?

Table 4 shows that the variety of aspects reported from SLR ranged from *problem solving*, *change*, *challenging* nature, *science* to the *experimental* aspect of the discipline.

Table 4: Motivational aspects of software engineering (Beecham et al 2007).

Motivating Aspects of software engineering field	%
Asp1: Problem Solving (the process of understanding and solving a problem in programming terms)	3
Asp2: Team Working	2
Asp3: Change	4
Asp4: Challenge (Software Engineering is a challenging profession and that in itself is motivating)	4
Asp5: Benefit (creating something that is of benefit to someone or enhances well-being)	3
Asp6: Science (making observations, identifying, describing, engineering, investigating and theorising, explaining a phenomena)	2
Asp7: Experiment (trying something new, experimentation in order to gain experience):	2
Asp8: Development practices (Object Orientated, XP and prototyping practices)	2
Asp9: Software process/lifecycle – Software development, project initiation and feasibility studies, and maintenance (note maintenance was also found a de-motivating activity)	1

4.5 What other Models of Motivation Exist in Software Engineering?

Table 5 provides a summary of the different models of motivation and job satisfaction developed specifically for the Software Engineering. It shows that almost all the models draw on and build upon some classic theory of motivation, however, none of the models substantially encapsulates all the factors that underpin motivation. In effect, none of the models reflects all the concepts of motivation expressed in the literature as presented in Section 2.3.

Table 5: Models for motivating SEs (Beecham et al 2007).

Models of motivation	**#**
1: Job Characteristics Model (JCM) of Software Engineer (SE) Motivation (development, enhancement or validation)	10
2: Models focusing on Software Engineer Job Satisfaction	6
3: Models of Open Source Developer SE Motivation	3
4: Models of leadership influence on SE motivation	3
5: Model drawing on expectancy theory, goal-setting theory, and organizational behaviour specific to the software development process	1
6: Model of Task Design influence on SE motivation	1
7: Model of Career Progression influence on SE motivation	1
8: Social support influence on Software Engineer turnover	1

Overall, we suggest that though these models above provide valuable insight into software engineers' motivation, they are disparate and do not encompass all the factors that our understanding of the literature leads us to expect of a model on motivation.

5 MODEL CONSTRUCTION PROCRESS

In Section 3, we presented the framework used to explore the issues in this research. This framework summarised the research strategy used to frame our research questions. From this framework, we were able to evolve a model of motivation, based on the answers to the Research Questions 1 to 4. We present our model of motivation as a series of stages in a process. In this model, for example, we classify a software engineer's motivation as two sets of factors: aspects inherent in the job that software engineers do, for example the problem solving nature of software engineering and some general factors, which can also be sub-categorised into extrinsic and intrinsic factors as illustrated in Fig. 2.

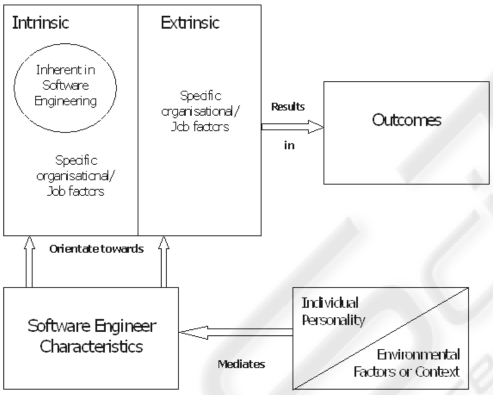
Intrinsic
Extrinsic
Inherent in Software Engineering
Specific organisational/ Job factors
Specific organisational/ Job factors
Results in
Outcomes
Orientate towards
Software Engineer Characteristics
Individual Personality
Environmental Factors or Context
Mediates

Figure 2: Model of motivation in software Engineering.

Our model also presents a set of characteristics of software engineers. It shows that these characteristics are mediated by individual personality traits and environmental factors. We suggest that these characteristics orientate software engineers towards their motivators, particularly, the set of inherent job characteristics. Our model shows that software engineers' motivation leads to external outcomes like staff retention, increased productivity and reduced absenteeism.

Having established the overall structure of the model, it can be parameterized with data from the SLR that we have presented in Section 4.

5.1 Refining the Model

A final stage in our model construction phase was to compare our model with the other models of motivation in software engineering in order to refine the structure and dynamics of our model. Our analysis of the structure of these other models showed that the relationships between components are more complex than Figure 2 suggests. For example, we were able to discern that contextual factors have a direct effect on motivators and how effective they are. It also became clear that the balance between organisational intrinsic and extrinsic motivators and the motivators inherent in software engineering have an effect on software engineers' characteristics, and their reactions to different motivators. The full discussion of these finding are presented in (Sharp et al 2007). The results of this analysis led us to refine our initial model. Figure 3 presents the refined model.

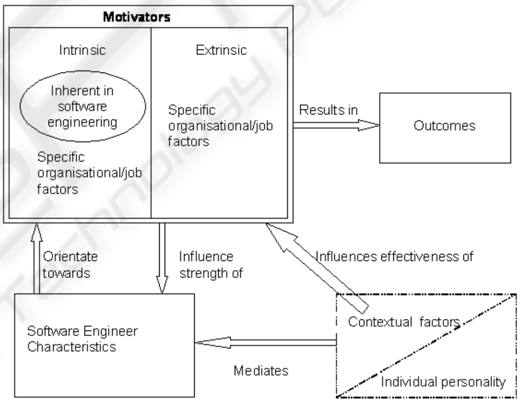
Motivators
Intrinsic
Extrinsic
Inherent in software engineering
Specific organisational/job factors
Specific organisational/job factors
Results in
Outcomes
Orientate towards
Influence strength of
Influences effectiveness of
Software Engineer Characteristics
Contextual factors
Individual personality
Mediates

Figure 3: Refined model of motivation in software engineering.

Overall, the model we present in Figure 3 is a process model because it explains the dynamics of motivation in a series of steps and stages as shown earlier. We suggest that this is so because the majority of studies in our SLR from which we have derived the above model, concentrated more on process theories than content theories. However, the notion of a process theory is not mutually exclusive from that of a content theory, so the model we have will also reflect some content theories of motivation as will be demonstrated next.

5.2 Theoretical Efficacy

The dynamics of the model presented here and the constituents of the components of the model strongly reflect three classic motivation theories of Job

Characteristics, Motivation Hygiene and Task Design. Our model also moderately reflects Adams' Equity theory, because the model parameters include the factor of equity as a motivator. We argue that even though our model fails to directly show a transition in needs or mix in needs, it identifies a set of motivators that can be said to reflect the Needs theories of Maslow and McClelland. The least reflected theory in our model is the Stimulus Response theory. Overall, we suggest that this strong reflection of many of the classic motivation theories in our model gives the model the theoretical efficacy that has been wanting in many models of software engineers' motivation.

6 CONCLUSIONS

In this paper we have presented a model of software engineers' motivators. Our previous work suggested that no rational model of software engineers' motivation existed. The few that exist make a fairly disjointed use of classic motivation theories. In this work, we have conducted a comprehensive review of studies on software engineers' motivators and extracted a model of motivation from this review.

Our model shows that software engineers are motivated by two sets of factors, intrinsic and extrinsic motivators, where a subset of intrinsic motivators are aspects inherent in the job that software engineers do, for example, the problem solving nature of software engineering. Our model also shows that factors that orientate software engineers toward these particular motivators are their characteristics and that these characteristics are mediated by individual personality traits and environmental factors. Our model shows that the environmental or contextual factors can have a direct effect on the effectiveness of motivators. Also, that the balance between organisational intrinsic and extrinsic motivators and the motivators inherent in software engineering have an effect on software engineers' characteristics, and their reactions to different motivators. Finally, our model shows that the external outcomes of software engineers' motivation are benefits like staff retention, increased productivity, and reduced absenteeism.

Overall, we suggest that our model serves as a valuable starting point for managers wanting to understand how to get the best out of software engineers, and an individual wanting to understand his own motivation, or who is embarking on career choice. We suggest that it also provides a platform from which subsequent researchers can base their empirical studies, thereby providing a well-founded basis on further motivation work.

ACKNOWLEDGEMENTS

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

REFERENCES

Adams, J. S., 1963. Towards an Understanding of Inequity, *Journal of Abnormal and Social Psychology*, 67, 422-436.

Beecham, S., Baddoo, N., Hall, T., Robinson, H. and Sharp H., 2006. Protocol of a Systematic Literature Review of Motivation in Software Engineering, *Technical Report No. 452* School of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire.

Beecham, S., N. Baddoo, T. Hall, H. Robinson, and H. Sharp 2007/8, Motivation in Software Engineering: A Systematic Literature Review. *Journal of Information and Software Technology*, Elsevier (accepted manuscript).

DeMarco T and Lister T., 1987. *Peopleware - Productive Projects And Teams*. Dorset House.

Hackman, J. R. and Oldham, G. R., 1976. *Motivation through the design of work: test of a theory*. New York, Academic Press.

Hall, T., S. Beecham, N. Baddoo, H. Robinson, and Sharp H., 2007, Motivation Theory in Software Engineering *ACM Transactions on Software Engineering and Methodology (TOSEM),* (accepted manuscript).

Kitchenham, B., 2004. Procedures for Performing Systematic Reviews, Keele University and National ICT Australia Ltd.**:** 1 - 28.

Locke, E. A., 1968. Toward A Theory Of Task Motivation And Incentives, Organisation Behaviour And Human Performance, 3, pp 157-189.

Maslow A., 1954. *Motivation And Personality*. Harper & Row, New York.

McClelland, D. C., 1961. *The achieving society*. Princeton: Van Nostrand.

Pidd, M., 1999. Just Modeling Through: A rough Guide To Modeling, *Interfaces*, 28(2), 118-132

Sharp, H., Baddoo, N., Beecham, S., Hall, T., and Robinson, H., 2007. Models of motivation in software engineering, *Information and Software Technology*, (accepted manuscript)

Skinner B. F., 1976. *Walden Two*. Macmillan, New York.

Standish, 1994. *The CHAOS Report*, The Standish Group

Tomayko, J. E. and Hazzan, O., 2004 *Human Aspects of Software Engineering,* Charles River Media,

Vroom, V. H.,1964., *Work And Motivation*, Wiley, New York.