ISSN 1744-1986

The Open University

Technical Report N^o 2007/03

The Motivation of Software Engineers: Developing a Rigorous and Usable Model

Tracy Hall
Nathan Baddoo
Sarah Beecham
Hugh Robinson
Helen Sharp

19th April 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

1. Introduction

This report presents a summary of the work undertaken in the one year EPSRC “Modelling Motivation in Software Engineering” (MoMSE) project (2005). The aim of this work is to produce a model of motivation in software engineering. We give an overview of how we developed a model of motivation in Software Engineering (SE). Our model of motivation reflects three viewpoints: 1. motivation in the SE literature; 2. classic theory of motivation to include models tailored specifically to reflect motivation in software engineering; and 3. empirical investigations into the motivation phenomenon. The three viewpoints are represented in our model of motivation as follows:

1.1 First-cut model/The SE literature on Motivation. Our initial model of motivation is based on a systematic review of all the literature in the area dating from 1980 to mid 2006. A rigorous selection process resulted in a final set of 92 papers. A synthesis of themes taken from the literature underpins our first cut model of motivation (Beecham et al 2006; Beecham et al 2007a (in review)) This first model provides a basic structure and sets the initial parameters.

1.2 Second-cut model/Classic motivation theories and SE models of Motivation. This model is also based on the literature. However in this case we look outside of the SE literature to examine how our model, and other models of SE (in the SE literature) compare to theories derived from social scientists (Hall et al 2007 (in review)). We use the classic theory literature to enhance and validate our first cut model (Baddoo et al 2007 (in review)). We consider overlaps and how we might learn from classic motivation theories developed by people like Herzberg, Maslow and McClelland. We look at how software engineering researchers have drawn on the classic theories in their models of motivation, and where they have presented a new approach to the phenomenon. We re-examine the SE literature on models of motivation to guide our model development (Sharp et al 2007a (in review)). We combine the findings of these two strands of work to create our second cut model.

1.3 Model validation/Empirical studies. We validate our model derived from the vast body of literature shown in 1.1 and 1.2 above by conducting some primary studies. Our empirical studies comprise several datasets (observational data/interview data/workshop results) that directly observe how practitioners appear to be motivated in practice. This primary data was collected by members of the project team prior to and during the one year project. The aim of this work is to take a fresh and current view of the phenomenon and to validate the first and second cut models with these new findings. The two empirical studies looked at:

(a) motivation, software engineering tasks and environment: this study examined how practitioners are motivated by interactions with their peers (Beecham et al 2007b (in review)). We considered the strengths and weaknesses of the XP environment in terms of how this motivated software engineers. The data collections for this study were undertaken prior to the project. The data analysis was performed during the project. Our analysis is based on two data sets: (i) notes taken while observing how practitioners in six different XP companies develop their software; (ii) semi-structured interview data with 9 practitioners at a level 5 software development organisation practicing traditional development methods. We compared and contrasted the two data sets to see how methods and behaviour of the XP teams matched what the traditional environment practitioners felt contributed to project success. This exercise highlighted factors that we need to consider for inclusion in our second-cut model of motivation.

(b) motivation and different roles in software engineering: A separate study was conducted during the project to examine how practitioners in different roles perceive they are motivated (Sharp et al 2007b (in review)). This data was collected at a workshop specifically designed to explore how project managers and developers feel they are motivated; what is important to each of these groups; whether they form a homogeneous group with differing needs or whether they have similar needs.

Figure 1 shows the stages involved in developing a rigorous, usable model of motivation:

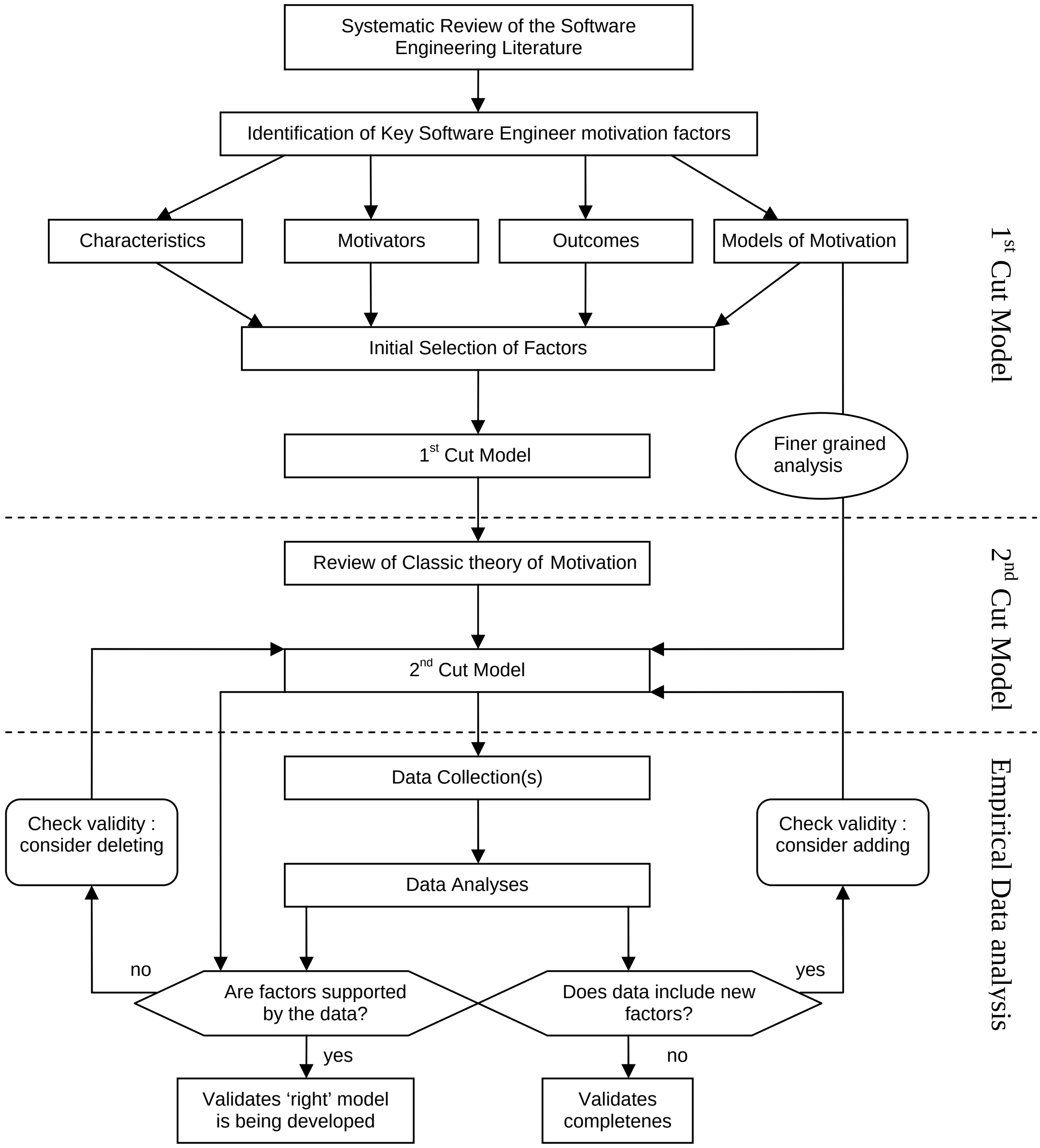

Figure 1: Model development process (adapted from (Dybå 2000))

The rest of this report is organised as follows: In Section 2 we present an overview of the first cut model derived from our systematic literature review, in Section 3 we describe how we used theories and model literature to refine our first cut model. In Section 4 we show how the empirical data was used to validate and propose potential refinements for our model. In Section 5 we conclude and reflect on future work.

2. Overview of the First Cut model

Our first cut model is derived from our systematic literature review (SLR) of motivation in software engineering ((Beecham et al 2006; Beecham et al 2007a (in review)). The aim of the SLR was to identify and bring together previously published work addressing the following research questions:

RQ1: What are the characteristics of Software Engineers?
RQ2: What motivates and de-motivates Software Engineers to be more or less productive?
RQ3: What aspects of Software Engineering motivate and de-motivate Software Engineers?
RQ4: What are the external signs of motivated and de-motivated Software Engineers?
RQ5: What models of motivation exist in Software Engineering?

In this section we provide a very brief overview of how we collected the SLR papers and identified the factors that populate our first cut model.

2.1 Performing a systematic literature review

We used the following systematic review steps (Kitchenham 2004)[1]:

1. Identify the need for a systematic literature review (MoMSE(cfs) 2005)
2. Formulate research question(s)
3. Carry out a comprehensive, exhaustive search for primary studies
4. Assess and record the quality of included studies
5. Classify data in terms of answering the research question(s)
6. Extract data from each included study
7. Summarise and synthesise study results (meta-analysis)
8. Interpret results to determine their applicability
9. Write-up study as a report

Searching for primary studies. Key words are identified for each research question. We used these key words to search 8 popular academic databases as well as key conference proceedings, journals and authors. Our searches elicited over 2,000 references. Evaluating the title and abstract enabled us to reject approximately 1,500 of these. We then looked at 519 papers in full to establish a final list of 92 papers.

Including and excluding primary studies. Any published work that directly answered our research questions and was published between 1980–2006 was considered for inclusion in our review. To be included, the study must also be published in a journal paper, conference proceedings, or empirical experience report based on theoretical or previous rigorous research.

[1] Full detail of these steps are provided in our protocol 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. 453* School of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire. which can be downloaded from:
http://homepages.feis.herts.ac.uk/~ssrg/MOMSEProto.htm.

Assessing the quality of primary studies. Each included study was assessed against a quality checklist. Scores were given according to whether the study presented clear, unambiguous findings based on evidence and argument. Scores for the 92 papers resulted in 82% of papers scoring good, very good or excellent.

Extracting and synthesizing data from included primary studies. We used Endnote version 9 (www.endnote.com) to record reference details for each study. How each study answers the research question(s) was recorded on a separate results form. We synthesised the data by identifying themes emanating from the findings reported in each accepted paper.

Data Validation. We performed two validation exercises. First we ran an inter-rater reliability test on the 519 paper references we found in our initial search. A 99.4% agreement was recorded with the original assessments. Second we performed a final validation exercise using an independent expert on the 95 'accepted papers'. Again there was a high level of agreement between the primary researchers and the independent expert (99.8%). Three of the accepted papers were rejected as a result of this exercise, leaving 92 papers for inclusion.

2.2 Model Creation

We used the research question to create an initial framework for our model as shown in Figure 2:

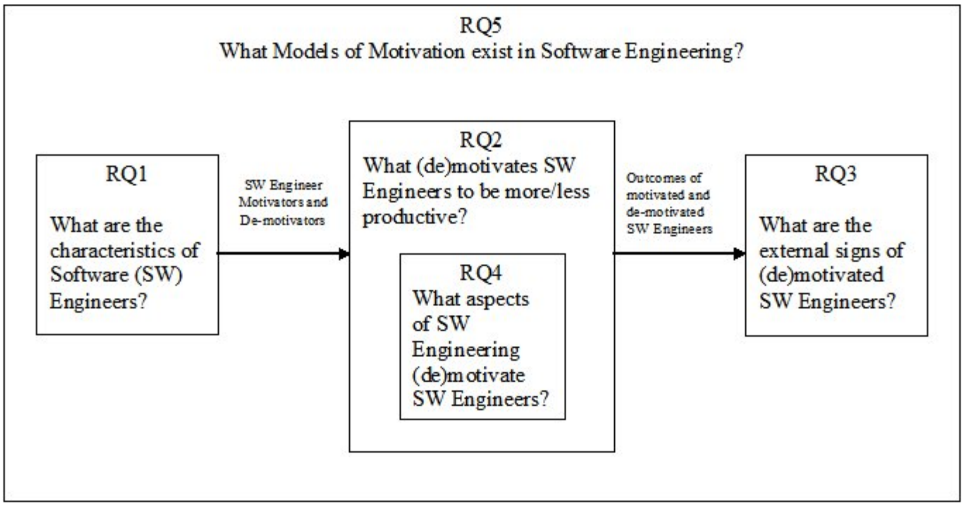

Figure 2: The relationship between our five Research Questions

We then populated the framework suggested by Figure 2 with factors derived from a data synthesis of the SLR findings. This first cut model is shown in Figure 3. A copy of the SLR paper (Beecham et al 2007a (in review)) along with all other papers produced as part of this project can be sourced at http://homepages.feis.herts.ac.uk/~ssrg/MoMSE.htm .

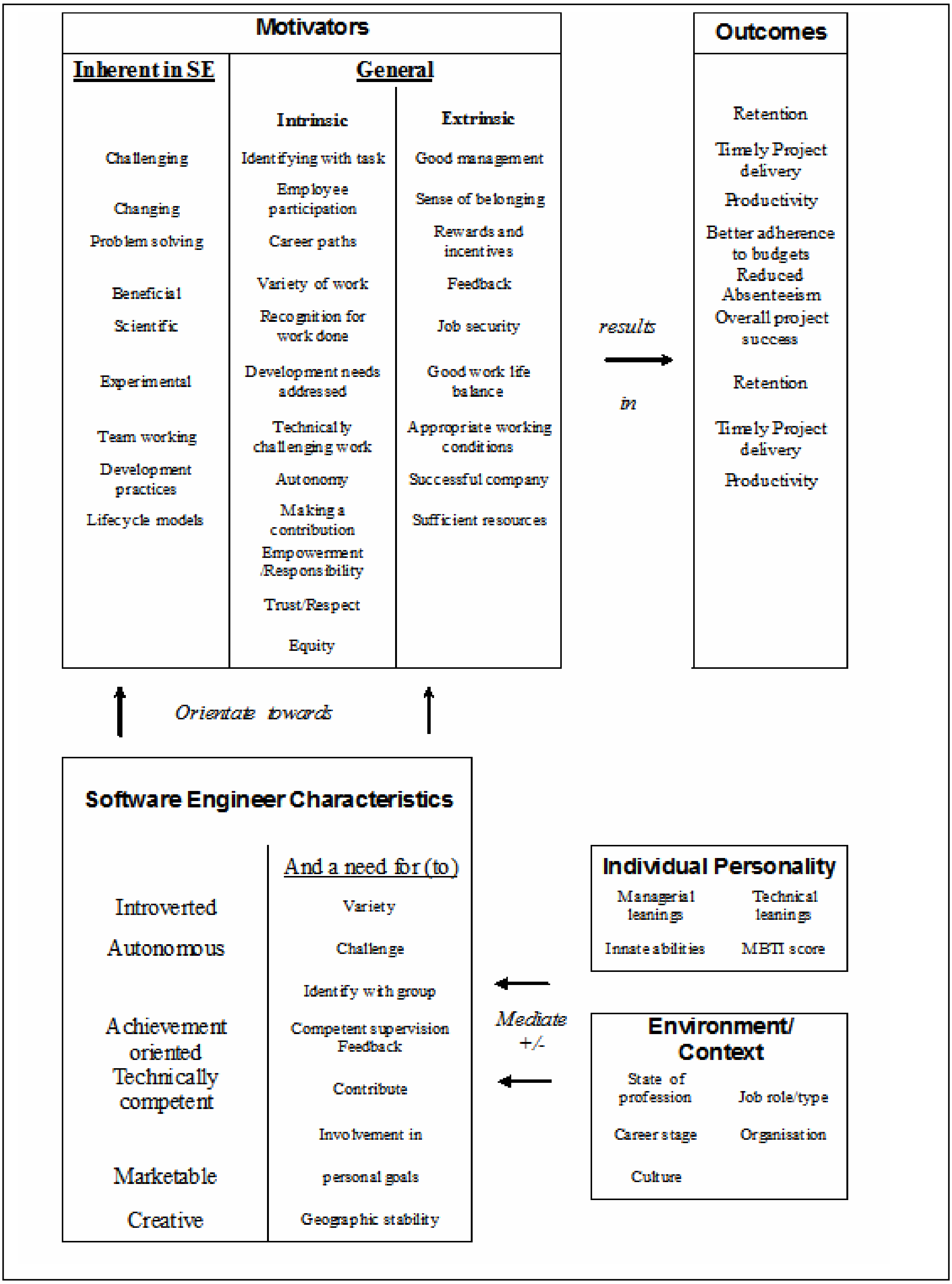

Figure 3: First Cut Model of Motivation – From Systematic Literature Review Results

3. Overview of our second cut model

The second cut model emerged from a further examination of the models of motivation in SE (Sharp et al 2007a (in review)). The SLR studies on which our first and second cut models are based were placed in context with classic theories of motivation drawn from the social science literature in (Hall et al 2007); and in a validation exercise we employ theory to gain a better understanding of our view of SE motivation outlined in our first cut model (Baddoo et al 2007 (in review)).

In our examination of the classic theories of motivation we identified the distinct contribution made by eight theories (Hall et al 2007). We then re-examined all the papers collected in our SLR to establish the level of theory use in each study. We found that over half the studies did draw on a classic theory of motivation, 15 papers draw on other theories, while 26 papers did not explicitly draw on any theory at all. We identified that our first cut model resembles the Job Characteristics Model, and considered how the other seven theories might feed into our model.

Development of our second cut model is explained in our study of models of motivation (Sharp et al 2007a (in review)). This new version of the model includes changes based on other models of SE motivation where relationships and structure of model components are developed. It is clear that existing models either rely heavily on one model (the job characteristics model), or are quite disparate and difficult to combine. Baddoo et al (2007) takes a close look at the factors and parameters of the first cut model that arise from our research questions 1-4. This study uses theory to validate the first cut model. This validation supports our aim to capture the current state of understanding of motivation in software engineering. Baddoo et al (2007) also explains the model construction process that results in the model shown in Figures 4-6.

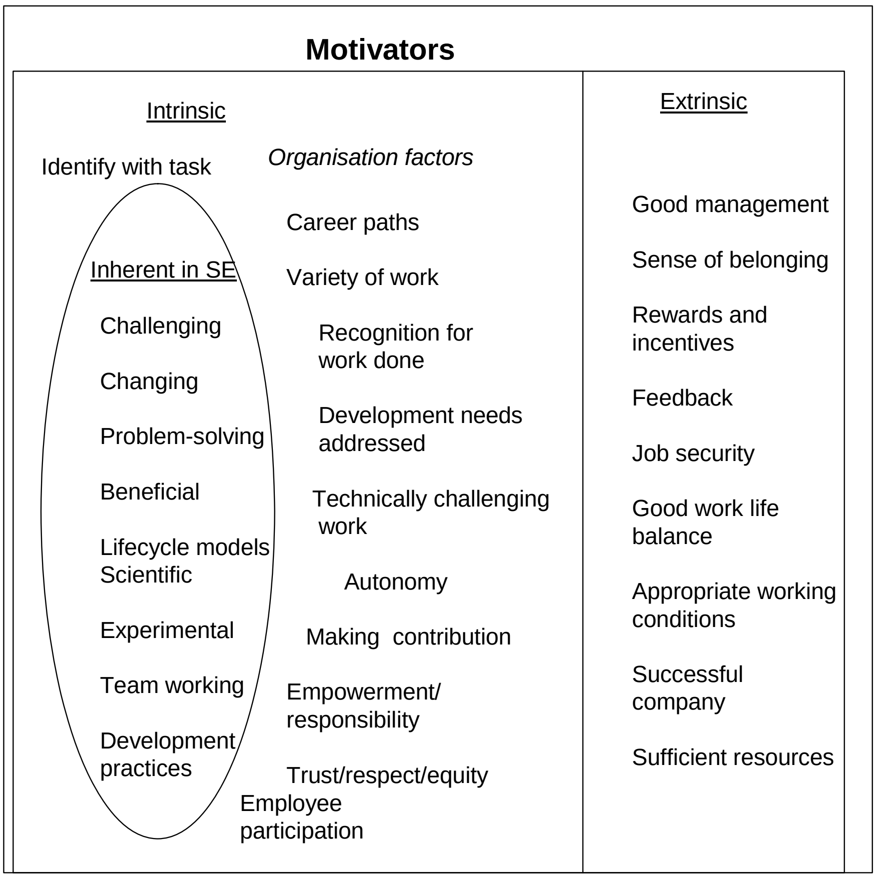

Figure 4: Second Cut Model of Motivation parameters

Figure 4 shows the importance of intrinsic motivations which are the motivators that relate directly to the task. This is a development from our first cut model.

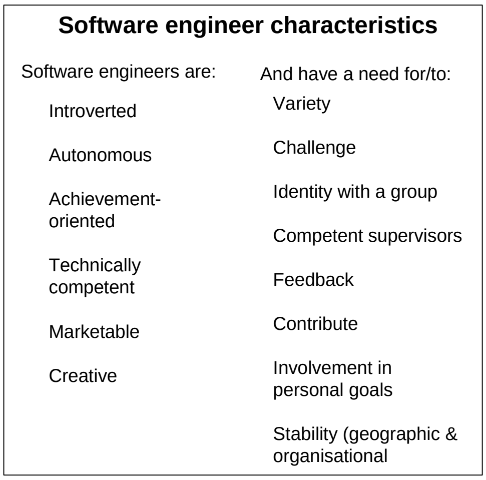

Figure 5: Second cut model Software engineer characteristics parameters

Figure 5 shows the parameters of our evolving software engineer characteristics, that answered our original research question 1 in our systematic literature review remain largely unchanged. The main change to our original model comes in the framework, where we see that there are different relationships between model components. For example, contextual factors have a direct effect on motivators and how effective they are. It is also clear that the balance between organizational 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 model shown in Figure 6 represents our second cut model that takes these factors into account.

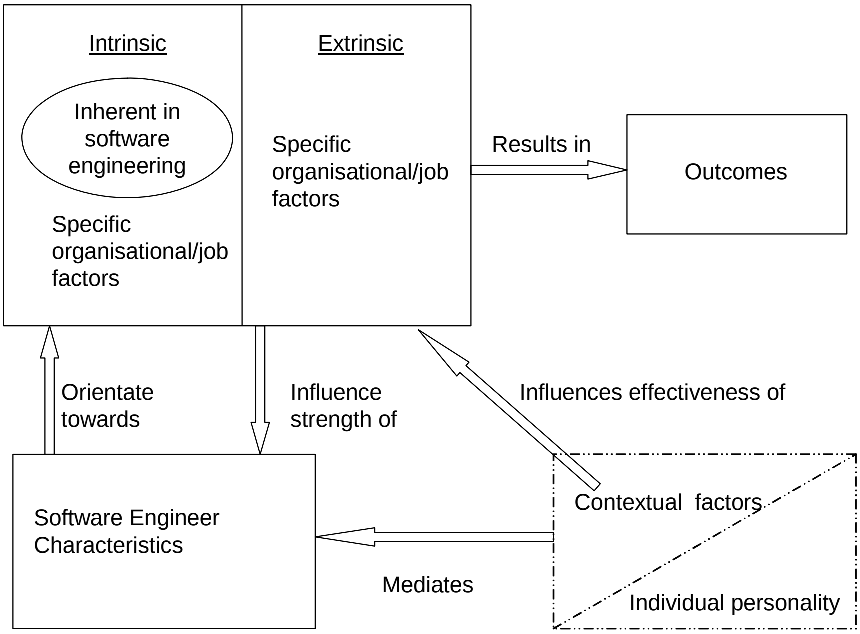

Figure 6: Second Cut model – Framework (Sharp et al 2007a)

One of the key findings from analyzing the literature for questions 1 – 4 is that motivation is heavily dependent on context. We represent this in our model by the split component 'Individual Personality' and 'Environment'. The impact of context is more complex than we initially suggested, but it is also clear that the notion of 'context' itself has several layers each with its own impact on motivation. The literature on models does not shed much light on how this influence works, but we identify layers shown in Figure 7.

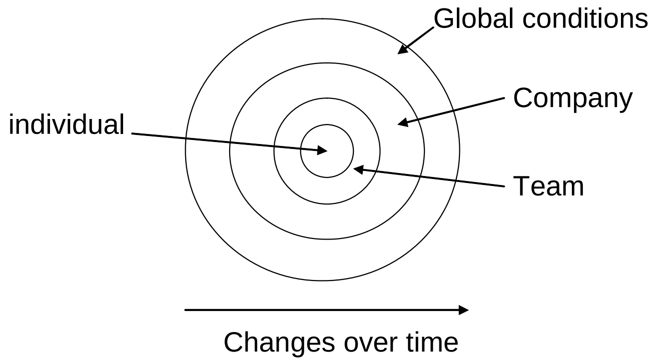

Figure 7: The spiral of contextual layers

4. Model Validation

Validation is the substantiation that the components within the model's domain of application possess a satisfactory range of accuracy consistent with the intended application of the model (Sargent 2000). Carson and Robinson's definitions are pertinent to this study where validation is defined as "the process of ensuring that the model is sufficiently accurate for the purpose at hand" (Carson 1986), or whether the right model is being built (Robinson 1997).

In this report we validate whether the right model is being built by looking at what practitioners find motivating, as abstracted from interview data from 9 software engineers (Beecham et al 2007b (in review)). Issues that these practitioners found important to supporting them to produce good quality software are noted in Table 1.

Table 1: General themes emerging from the interview data (both negative & positive)

Cross-cutting themes	Definition
progress	Relates to movement towards and achievement of goals, where goals might be milestones, meeting requirements, quality, timescales
access to information	Relates both to the flow of information regarding development matters, and the desire of individuals to acquire more information and to learn
openness/ communication	Relates to the flow of information, but it focuses more on the culture of the team and of individuals' willingness to help others, to say what they believe and to encourage others to do the same
responsibility/ autonomy/ ownership	Relates to Software Engineers defending their best ideas, voicing problems, working exceptionally hard, a culture where people don't blame others. Not waiting to be told what to do. Ownership of a process and piece of code.
software quality	Covers software reliability, integration, dependencies, complexity, meeting requirements, design, defects, latent faults, maintainability, solutions
team morale	Includes (de)motivation, team dynamics, drive others in team, morale boosting, blame culture, punitive management practices, negative effect on team, friction, team cohesion, annoying others, difficulty working with people.
fear/insecurity/ confidence	Includes both positive and negative characteristics. Behaviour is influenced by encouragement (or lack of it); threats from other team members performing too well; management personalising punishment for not meeting targets; treating others with distain. Job security. Behaviour is defensive and paranoid. Can be over-confident, which is seen as negative.

We take these cross-cutting themes and consider whether our model of motivation captures these needs. This direct analysis is given in table 2.

Table 2: Validation of MoMSE Model of Software Engineer Motivation		
Cross-Cutting themes from (Beecham et al 2007b)	**Motivators in Model**	**Potential model refinement**
progress	Career path/challenging task/recognition for work done/good management/rewards incentives	**Goal-setting (**also a theoretical model)
access to information	Development needs addressed Sufficient resources	
openness/ communication	Employee participation/making a contribution	
responsibility/ autonomy/ ownership	Trust/respect Empowerment/responsibility	
software quality	Recognition for work, technically challenging work, problem solving	Producing quality work
team morale	Team working, sense of belonging	
fear/insecurity/ confidence	Job Security, trust, respect	

Our model addresses all the issues raised, however there are some candidate improvements that we could explore in future work, such as goal-setting (shown also to be important in our theories paper – Hall et al 2007), and producing quality work.

We also look at software engineer characteristics as an integral part of our model (factors of which are given in figure 5). Some of the empirical data findings also link to these characteristics. Technical competence is a characteristic of a software engineer, which is matched by the need to produce high quality software. Another recognised characteristic of a software engineer is that they tend to be more comfortable working in an autonomous manner, which matches with our cross cutting theme of autonomy. The fear/insecurity/confidence cross cutting theme is likely to be due to the software engineers need for security, which is another recognised characteristic.

This validation exercise, although from a small group of practitioners and is therefore not necessarily representative, does indicate that the model is also usable in practice. Indeed, at a workshop (Sharp and Hall 2007) we presented this model to practitioners which sparked a significant debate on motivation indicating that the model was regarded as relevant by participants. It also shows that looking at both motivators and characteristics of the software engineer helps to give a broader understanding of the phenomenon. However we rely on future work to confirm this empirically.

5. Conclusion and Future work

The output from this feasibility study (MoMSE 2005) is a model of motivation that is specifically tailored to the software engineering field today. In this report we have shown how we have taken a rigorous approach to creating a framework and populating the framework with factors that relate directly to practitioners in software engineering. The model draws on previous models suggested in the literature (some of which have been used in practice), leading us to believe that our model should also be usable. However we rely on future work to confirm this.

Future work may take a variety of forms. Data may be collected through questionnaire or interview surveys, underpinned by rigorous empirical observation and engagement with individual software practitioners. Results of this further investigation will lead to further refinement of this model. Having produced a refined model we will then need to conduct a validation exercise.

Once we have developed a refined model based on further empirical studies, we will verify and evaluate the model's usability. Verification is defined as the process of ensuring that the model design (conceptual model) has been transformed with sufficient accuracy (Davis 1992), testing whether the model is built correctly (Robinson 1997), and ensuring that the model components are correct (Sargent 2000).

Evaluation, however, encompasses both validation and verification activities along with the model's quality, usability and utility assessment (Gass 1983). Our future work may involve an evaluation of the model. Results of our evaluation will help with further model development, where we envisage developing the model iteratively.

Acknowledgement

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

References

Baddoo, N., Hall, T., Beecham, S., Robinson, H. and Sharp, H. (2007 (in review)). Motivating Software Engineers: A theoretically reflective model *Journal of Systems and Software*.

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. 453* School of Computer Science, Faculty of Engineering and Information Sciences, University of Hertfordshire.

Beecham, S., Baddoo, N., Hall, T., Robinson, H. and Sharp, H. (2007a (in review)). Motivation in Software Engineering: A Systematic Literature Review, *ACM Computing Surveys*.

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

Carson, J. S. (1986). Convincing Users of Model's validity is Challenging Aspect of Modeller's Job, *Industrial Engineering* 18 (6): 74-85.

Davis, P. (1992). Generalizing Concepts of Verification, Validation and Accreditation for Military Simlation, Rand, Sant Monica, CA.

Dybå, T. (2000). An Instrument For Measuring The Key Factors Of Success in Software Process Improvement, *Empirical Software Engineering* 5 (4): 357-390.

Gass, S. (1983). Decision aiding models: Validation, Assessment, and Related Issues for Policy Analysis, *Operations Research* 31 (4).

Hall, T., Baddoo, N., Beecham, S., Robinson, H. and Sharp, H. (2007 (in review)). Motivation Theory in Software Engineering *IEEE Transactions on Software Engineering*.

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

Momse(Cfs) (2005). Modelling motivation in software engineering *Case for Support.* (EPSRC Proposal, EPSRC Reference: EP/D057272/1).

Robinson, S. (1997). Simulation Model Verification and Validation: Increasing the User's Confidence. *Proceedings of the 1997 Winter Simulation Conference*.

Sargent, R. (2000). Verification, Validation, and Accreditation of Simulation Models. *Winter Simulation Conference*.

Sharp, H., Baddoo, N., Beecham, S., Hall, T. and Robinson, H. (2007a (in review)). Models of motivation in software engineering *IST*.

Sharp, H. and Hall, T. (2007). What motivates software practitioners?, *Workshop presented at SPA 2007, Cambridge 25-28 March*.

Sharp, H., Hall, T., Baddoo, N. and Beecham, S. (2007b (in review)). Exploring Motivational Differences between Software Developers and Project Managers, *ESEC/FSE*.