# Effort Estimation of Software Development Projects with Neural Networks

Diplomarbeit

zur Erlangung des Grades eines Diplom-Ökonomen der Wirtschaftswissenschaftlichen Fakultät der Leibniz Universität Hannover


vorgelegt von

Name: Petersen

Vorname: Marten

[redacted]

[redacted]


Erstprüfer: Prof. Dr. Michael H. Breitner

Hannover, den 28.09.2007

**Table of Contents**

## 1. Introduction: Business Value of Software Metrics

Today's business world is increasingly shaped by globalisation, virtualisation and business networks. Whereas in 2007 only 2% of the Gross Domestic Product [GDP] are to be generated in the "project economy" this figure is expected to rise to 15% by 2020[1]. The ubiquitous presence of high-speed networks and technological sophistication enables dynamism and flexibility and demands adaptable and highly interconnected software applications. Software due to its cross sectional character becomes a means to an end as opposed to an end in itself and therefore an innovation enabler. In fact software being part of products in the service consumption and investment economy has become an invisible technology to the consumer.

In this highly dynamic and fast moving business world the reliance of business processes on information technology [IT] increases. IT itself enables interconnectivity and seamless processes between and within organisational units. The need for new or adapted software as the facilitator of information collection, processing and transfer is therefore greater than ever, whilst the underlying complexity of systems increases.

One of the challenges of the software industry is the efficient allocation of resources in an effective manner. As the software industry works merely project based, the challenge is to complete projects within time and budget as well as meet the functionality and quality demanded by the customer. There is an apparent trade-off but also a correlation between these dimensions which practitioners usually call the magic triangle of project management[2], i.e. a trade-off between time to deployment and development costs.

Some call the creation of software an art in itself, others a rather mechanical task. Whichever view one takes, conducting a software development project [SDP] and producing an immaterial good cannot or at least only to a very limited extent be automated. Mental work is required and consequently makes up for the majority of incurred costs. Obtaining knowledge of the required resources as early on as possible in the project lifecycle is critical to success. Therefore effort estimates and consequently cost estimates of SDP's gain a strategic role in a range of situations for the organisation or the project itself. For the organisation, the allocation of the resource pool is a crucial task as a high utilisation enables high revenue. Estimates can furthermore act as a catalyst for innovation, as budgets can be allocated towards the more productive or more profitable project teams, profit centres or the like. Projects benefit of the estimates as a tool to justify the use of those resources in the

---

[1] Cf. Jan Hofmann (2007).
[2] Cf. Poensgen (2005) p. 675.

course of the project and as a measuring stick for success. An estimate based on software metrics as explored in this thesis can furthermore point out the trade-off between time, cost, quality and scope also known as the magic square of project management. In a bid situation, estimates act as a risk reduction for the bidding strategy as an overpriced bid results in a loss of the contract and bidding too low means losing profitability for the organisation.

Estimates also have psychological implications. Following "Brooks' Law" which was stated by Fred Brooks in his book "The Mythical Man-Month"[3] additional manpower in a project that is likely to run out of schedule will itself have a negative impact on the timeliness. The reasoning behind this hypothesis is that new workers have to familiarise themselves with the project and that the need and allocated time for communication has to increase. Overestimating effort has equally negative effects on the performance of the project. "Parkinson's Law" states that "work expands so as to fill the time available for its completion"[4]. People involved in a project will stretch their work by reducing their productivity in order to deliver in time. Therefore the estimate becomes a self fulfilling prophecy. Another effect of overestimates is called "gold-plating", i.e. adding unwanted or unnecessary functionality to the product.

Estimates can be further complicated by what is referred to as the estimation paradox. In the early phases of a project where an estimate is needed most, the prerequisites, i.e. underlying documents and specifications are lacking. Once it has been completed, the estimate can be made with absolute accuracy, i.e. effort figures are available and estimates become obsolete[5] [6].

Despite the crucial nature of estimates it is a field where the software industry has a huge potential for improvement. Surveys show that 60-80% of SDP are exceeding their budgets by 30-40%, mostly due to overly optimistic estimates[7]. There are a variety of methods to estimate effort, although the one consistent and appropriate method is yet to be found despite significant research over the last 30 years. Most organisations rely on expert based estimates and do not employ statistically founded and more objective methods. Methods are not

---

3 Cf. Brooks (2006).
4 Cf. Parkinson (1955).
5 Cf. Meli in Jones (2002) Early and Quick Function Point Analysis from Summary User Requirements to Project Management p. 427.
6 Cf. Ebert (1996).
7 Cf. Molokken, et al. (2003).

typically chosen from of a scientific rationale and therefore the business potential is not levied[8].

This thesis aims to outline the potential of a company specific effort algorithm using the neurosimulator FAUN[9] and data supplied by IBM Global Services [IBM GS]. In the second chapter, the author seeks to give a structured overview over the defining characteristic of a SDP and of what factors influence the effort of SDP's, with an emphasis on software metrics. Next, the most common methods of effort estimation are outlined including traditional statistical methods as well as especially Artificial Neural Networks [ANN]. Their suitability for different environments is evaluated on the basis of existing research. In the fourth chapter the project data from IBM GS is examined and an effort prediction algorithm is computed employing the software tool FAUN. Effort drivers are examined and quantified and related to existing research. Finally the author concludes his findings and shows directions of further research and a broader application of ANN's in Software Engineering [SE].

[8] Cf. Moloekken-Oestvold, et al. (2004).
[9] Cf. Breitner, et al. (2003).

## 5. Conclusions and Outlook: Effort Estimation Methodologies as a Key Factor in Successful Software Development

Software development continuous to be a vital productivity driver and enabler of cost reduction, efficiency and automation in the business world. In recent times the economy has fundamentally evolved in character from a traditionally structured economy to a dynamic network of flexibly interlinked actors. It is the case today that software is needed to support and facilitate this flexibility, and thus SDP's gain strategic importance in the process of a prospering economy.

Remarkably, SE is considered a science that has been compared to other engineering disciplines, yet has only recently received the attention needed to develop the best practices and examine the processes in software development. Whilst some still believe that programming is an art, others regard it as fully-fledged engineering discipline. However SDP's are a phenomenon that have the tendency to deliver less than promised, later than planned, using more budget than intended at a disappointingly low quality. One might blame the difficulty of translating business requirements into software concepts on the expectation-delivery gap, though the driver of effort and therefore costs lie in the interaction of a plurality of factors of project, organisation and people and can succinctly be summarised as generic complexity.

This thesis has given a structured overview of characteristics that altogether make up the aforementioned complexity. When examining the drivers of effort, it is inevitable to have a size measure to relate the effort to. Special attention is therefore paid to software metrics that prove to be structured and relatively reliable methods to size the intended product of an SDP. The three outlined methods are all ISO certified and are today broadly applied across the industry. However all of the described methods are limited in their applicability to information based software and none of them can be regarded as a valid measure for real-time or algorithm intensive systems. Moreover, not only are they not convertible into one another, but they all require interpretation of the counting rules to judge upon the diversity of software applications. There is no silver bullet in software size metrics, though the existing approaches are better than a blind flight. Whereas software size metrics give an indication of how large the end product will be, the effort spent on the way to get there is influenced by innumerable factors as the majority of a project is human brain work. Project-specific, organisation-specific and human factors are examined and related research is presented quantifying huge ranges of possible impacts on effort. These studies can only give indications of the relative importance of effort drivers and not the desired overarching picture.

Heuristic, parametric and data mining methods to estimate the effort of a project are then described and evaluated, revealing that there are considerable differences in the degree of detail, objectivity, required data and degree of sophistication of the algorithm. Heuristic methods have their strengths in the relatively low effort spent on the estimation and the high degree of human creativeness and experience involved. The parametric models try to reflect the complex reality of SDP's but are limited in their applicability to the environment and the software processes they are developed under. Furthermore the multiplicity of characteristics in these models can give the false impression of accuracy. Parametric models are a suitable supplement to heuristic methods if no larger project database is available and can be used for plausibility checks.

Once a sufficient project database of 50 - 200 SDP's, depending on the variability of the environment, has been collected, statistical analysis can lead to more accurate organisation specific solutions. In particular, LSR is widely available and integrated in standard software packages and has the advantage of being highly transparent. Nevertheless, the implications and limitations of multivariate LSR need to be understood. ANN's as the most important subsection of AI depict structures that are adapted by the human brain. The topology of an ANN allows for high complexity of the underlying phenomenon. Through optimisation or training, ANN's are able to learn the relationship of inputs and outputs and are able to cope with a high level of noise in the data. However a relatively large amount of data is necessary for the training and validation of ANN's which only very few companies are able to obtain/ acquire. Furthermore the knowledge inherent in an ANN cannot easily be extracted, which raises problems when applying them to a business environment.

The comparison of the outlined methods on the basis of existing research shows that there is no single best methodology for effort estimation. The development of company specific algorithms is however favourable over all other methods.

The author was delighted to gain access to the BMRS of IBM GS, a database containing over 1500 SDP's with a large amount of project characteristics describing effort, duration, topology, quality and other factors of the project. The effort drivers not represented in this database are the human factors. Despite the relevance of experience, skill as well as other human related characteristics, the analysis is conducted without these. A structured analysis of the data with over 2000 training runs reveals that despite the large datasets, the prediction accuracy of the best ANN is comparatively low and it is advised to be used with caution. An in depth analysis shows that the ANN estimates projects with certain characteristics more accurately than others. The ANN can be used to quantify trade offs with respect to the choice of one of the characteristics, i.e. the choice of the primary database, and can thereby deliver business value. However the overall results are limited by the available characteristics and further research including human factors would be likely to increase the validity and integrity of the analysis.

This thesis has shown that the complexity of an SDP can only be accounted for through a similarly complex estimation methodology. The availability of project data limits the variety of methods that can be applied; although company specific estimation algorithms have the potential to outperform any other parametric method.

The panacea of effort estimation has not been found; neither the universal remedy. The scientific interest and diametrical findings in this field show that there is a lot of movement in this discipline. Nonetheless the author does not expect a breakthrough in research as company specific data repositories reflect confidential information about productivity and cost structures. It is therefore unlikely that other quality assured data repositories will be opened up for scientific research in the near future.

It seems that for an individual company, the effort to build up and maintain a data repository for effort estimation is unreasonably high in comparison to using Expert estimation or Expert Systems. A repository including software size metrics can be used in other business functions, such as planning and controlling as well as project controlling. Notably in project controlling, it enables methodologies that track and compare project and cost progression, i.e. the Earned Value Method.

For ANN's there is a wide spectrum of potential application and as information becomes instantly and available in overwhelming amounts, ANN's offer a suitable way of processing the information and extracting knowledge. Be it for example the prediction of share prices and their derivatives or the evaluation of medical studies, the applications are endless. Vital prerequisites for good results are a high quality data repository that includes all relevant factors of influence. Despite considerable attention in the academic society, training ANN's is still a try and error method and universally applicable methods still need to be developed.