ABSTRACT

Background: There has been much research into building formal (metrics-based) prediction systems with the aim of improving resource estimation and planning of software projects. However, the ‘objectivity’ of such systems is illusory in the sense that many inputs need themselves to be estimated by the software engineer.

Method: We review the uptake of past software project prediction research and identify relevant cognitive psychology research on expert behaviour. In particular we explore potential applications of recent metacognition research.

Results: We find the human aspect is largely ignored, despite the availability of many important results from cognitive psychology.

Conclusions: In order to increase the actual use of our metrics research e.g. effort prediction systems we need to have a more integrated view of how such research might be used and who might be using it. This leads to our belief that future research must be more holistic and inter-disciplinary.

Categories and Subject Descriptors
D.2.9 [Management]: Cost estimation; D.m [Miscellaneous]: Software Psychology

General Terms
Human Factors, Management, Measurement

Keywords
Software project management, effort prediction, accuracy, metacognition.

1. INTRODUCTION

Our starting point, to quote from the workshop call for papers is:

“the low impact that software metrics has on current software development”

Given that software engineering is an applied discipline, this is a worrisome state of affairs. Since the range of metrics and their applications is extremely diverse we choose to focus on a specific topic, namely project effort prediction.

There has been substantial research into building formal prediction systems because improved resource estimation and planning of software projects will have significant economic impact. However results using these methods are mixed and no single formal approach dominates. Furthermore, the objectivity of formal prediction systems is illusory in the sense that inputs, many of which are not objective measures, need themselves to be estimated by the software engineer. Therefore, we argue it is important to re-focus research energy on human aspects of prediction.

We explore how effort (usually as the major component of cost prediction) is an infrequent but high value decision. We show that it cannot be considered as a purely technical problem. We then go onto review research in cognitive psychology on expert performance and in particular human bias. Unfortunately, with a few exceptions, e.g. Jørgensen [14] and Rush and Roy [29] this has not been widely picked up on. In particular, whilst there is recognition of the problem of human bias (generally in the form of over-optimism and over-confidence), there is almost no work that considers de-biasing strategies, i.e. the practical steps that might be undertaken to improve matters.

2. BACKGROUND

2.1 Current Industry Practice

Flyvbjerg et al. [9] describe project cost underestimation and overrun as a global phenomenon which has not diminished in the last 70 years. They argue that “no learning seems to take place” (p.16). In addition, to project cost underestimation, benefits are overestimated. These phenomena are widespread and not restricted to software projects. Many studies attempt to quantify the cost of software failures. For example, the Standish Chaos Report (2004) found only 29% projects met their criteria for project success: projects on budget, on schedule, and with expected functionality and that the annual cost of cancelled projects was $55 billion. Whilst we might have methodological concerns with how the Standish figures are derived [17, 7], it is clearly indicative of extensive and persistent problems.

In terms of impact, Jeffery and Scott report “poor penetration in the application of empirical results” [13] in their review of the past 25 years of research. This is supported by the Moløkken and Jørgensen [25] review of industry sur-
The problem of obtaining useful predictions is compounded by the strong tendency for professionals to display over-optimism e.g. Buehler et al. [4] and over-confidence [15]. This leads to suboptimal project management and project outcomes. Because these phenomena are so widespread the causes of bias have been extensively investigated by cognitive psychologists in various domains over the past three decades since the seminal work of Kahneman and Tversky [20].

One problem is the so-called “planning fallacy”, the tendency to underestimate project completion times. Buehler et al. [4] examined the underlying cognitive processes and found that a narrow focus on future plans for the target task led to neglect of other useful sources of information. In other words, an illusion of control leads to significant over-optimism. Another source of bias is a preference for case-specific (and recent) evidence over distributional evidence [11]. This helps us understand why professionals struggle to learn lessons from the past. A related phenomenon is the peak-end rule where the most recent experience dominates even when it is highly atypical [19]. The dual-process theory of cognition leads to a tendency to trust analytic justifications (explanations) over intuitive ones yet to prefer intuitive judgements over analytic ones. One implication is that formal prediction systems can turn into “expert judgement in disguise” [16].

These biases are common to many problem domains and seem independent of individual differences (e.g. the traits of optimism and procrastination [3]). The limited work investigating de-biasing strategies, e.g. utilising previous experience, such as past project databases, Personal Software Process [12] and lessons learned or post mortem sessions, have not been successful, particularly in the field of software engineering prediction [21, 10, 15].

Evidence highlights both theoretical and empirical reasons for making consistently sub-optimal predictions within software engineering. However, the vast bulk of this research has been conducted using student participants working on problems that are not industry-related [24]. In addition, the literature has focused upon understanding factors that contribute to bias. We wish to move on and explore factors that promote de-biasing in realistic settings. In parallel, much research has been undertaken into metacognition, (that is knowing about knowing), particularly in the domain of education. There is a considerable body of evidence showing that increased metacognitive awareness leads to increased learning and enhanced performance e.g. [5]. In other words, metacognitive skills can be taught [2].

Metacognition can be divided into metacognitive knowledge and metacognitive skills. The former relates to declarative knowledge of the interactions among self, task, and strategy characteristics [8] which can be inaccurate and resistant to change. Clearly this will be an inhibitor to improving prediction performance. Metacognitive skills on the other hand refer to procedural knowledge for self-regulating problem solving and learning activities and include feedback (reflection) on metacognitive knowledge. This division between metacognitive knowledge and skills is related to that of single and double loop learning [1]. ‘Single-loop learning’ occurs when goals, values, plans and rules are taken for granted and put into operation rather than questioned. It reduces risk and affords greater control, but severely limits growth and learning. In contrast, ‘double-loop learning’ involves questioning the fundamental systems which underlie goals and strategies. It results in the questioning of governing variables and may lead to fundamental changes. This double-loop learning is necessary if practitioners and organisations are to make informed decisions in changing and uncertain contexts.
Reflection, a metacognitive skill important for personal and professional development [27]. It plays a key role in both single and double loop learning. However, critical reflection, as demonstrated in double loop learning is essential for growth and change. Critical reflection demands focusing on the cognitive aspects and challenging the strategies that led to particular actions, and the outcomes and lessons learned from those actions for future application.

Our previous studies of software project cost prediction suggest that feedback on performance and the typical methods for reflecting on experience, e.g. unaided lessons learned sessions, do not lead to improvement in accuracy or assessment of the uncertainty [21]. The lack of training in both reflecting on one’s own thinking and the fundamental causes of suboptimal outcomes (double-loop learning) can be a major obstacle. As an illustration, in a previous study where software professionals described reasons for their estimation errors [26] most were shallow and corresponded to single-loop learning. This lack of double-loop learning is a key contributor to the robust findings on over-optimism and over-confidence among software developers.

Mair has been investigating the relationship between metacognitive skills and performance amongst students [23]. Students reported feeling empowered and their learning was enhanced after using a software system that supported self-reflection. There is an exciting opportunity to combine these two research threads to explore techniques to help experts overcome problems and biases. Therefore our goal is to explicitly study expert prediction behaviour and how enhanced metacognitive awareness can improve performance.

3. DISCUSSION

We propose that in order to promote effective use of software metrics our research must be underpinned by theories of metacognition and double-loop learning from cognitive psychology. Specifically we intend to conduct experiments with software professionals to determine the impact of enhanced metacognitive awareness on the ability to improve project cost prediction and confidence (uncertainty assessment) within a software engineering context. To summarise, our thesis is:

1. Accurate prediction, e.g. of development effort, and uncertainty assessment is important for successful software engineering.

2. Formal prediction systems are not consistently reliable. Moreover their inputs and parameters must be manipulated by humans with a consequent loss of their raison d’être, objectivity.

3. There is a strong tendency for professionals to display over-optimism and over-confidence. A number of experiments and empirical studies help us to understand the cognitive basis for this bias.

4. De-biasing strategies based upon utilising previous experiences, such as lessons learned sessions, have not led to noticeable improvement in prediction accuracy or the realism of uncertainty assessment.

5. There are opportunities to use recent results from metacognition research to counteract this natural bias and consequently improve performance.

Although metacognitive skills can be taught [2], these are not necessarily automatically transferred to another context [6] so we need to design context-specific materials for use by software professionals. The aim must be to help software engineers challenge their biases which contaminate prediction processes and help them develop new insights into their causes. In order to accomplish this we need experiments to investigate the impact of different de-biasing strategies on software professionals in realistic settings and such experiments will need to be conducted by multi-disciplinary teams who include cognitive psychologists as well as computer scientists.

To clarify, confidence is defined as the reciprocal of the size of the estimate range i.e. the upper bound minus the lower bound1. This is distinct, but related, to estimation inaccuracy which is either zero if the true value is contained by the bounds or the distance from the nearest bound to the true value (a specialised form of absolute residual). Bias, for example over-optimism, is the mean direction of error, so the difference between the means of the predicted and actual values.

That metacognitive awareness increases confidence is not in question. However, achieving the level of confidence which leads to accurate, precise predictions is elusive. There is a danger that over-confidence leads to predictions based on unrealistically narrow ranges and consequently inaccuracy. Conversely under-confidence leads to estimates which are unworkable as the range of predictions is too broad e.g. the true value falls within the bounds of 1 to 100,000 person hours but this is hardly useful.

One of the reasons we believe double loop learning may be an effective de-biasing strategy is that it affords individuals and organisations the opportunity to engage in self-reflective learning by explicitly confronting assumptions and creating new more appropriate routines. Time for reflection is also essential in order to transform tacit experience into explicit knowledge [1]. Single-loop learning might endeavour to close the gap between predicted and actual by strategies such as “trying harder” or beliefs such as “this time we were unlucky”. Double-loop learning might ask why have I consistently under-estimated for the last ten projects?2?

It is essential to include the perspective of how people actually employ software metrics if our research is to have significant impact upon software engineering practice. A good example is research directed to enhance metacognitive awareness in professionals responsible for predicting and using software metrics. In doing so, our research can have the critical impact upon software engineering practice that we all desire.

4. REFERENCES


[3] R. Buehler and D. Griffin. Planning, personality, and prediction: The role of future focus in optimistic time  

1Strictly speaking this is a simplification since one might express confidence in the form of “I am 90% certain the true value will fall in the range $y_{n-1} \leq y_{new}$” which could be validated by observing multiple predictions.


