Information and Software Technology 53 (2011) 1382–1390

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation

Magne Jørgensen

Simula Research Laboratory & Institute of Informatics, University of Oslo, Norway

ARTICLE INFO

Article history:
Received 20 May 2010
Received in revised form 29 June 2011
Accepted 1 July 2011
Available online 23 July 2011

Keywords:
Effort estimation
Expert estimation
Human judgment

ABSTRACT

Context: The effort estimates of software development work are on average too low. A possible reason for this tendency is that software developers, perhaps unconsciously, assume ideal conditions when they estimate the most likely use of effort. In this article, we propose and evaluate a two-step estimation process that may induce more awareness of the difference between idealistic and realistic conditions and as a consequence more realistic effort estimates. The proposed process differs from traditional judgment-based estimation processes in that it starts with an effort estimation that assumes ideal conditions before the most likely use of effort is estimated.
Objective: The objective of the paper is to examine the potential of the proposed method to induce more realism in the judgment-based estimates of work effort.
Method: Three experiments with software professionals as participants were completed. In all three experiments there was one group of participants which followed the proposed and another group which followed the traditional estimation process. In one of the experiments there was an additional group which started with a probabilistically defined estimate of minimum effort before estimating the most likely effort.
Results: We found, in all three experiments, that estimation of most likely effort seems to assume rather idealistic assumptions and that the use of the proposed process seems to yield more realistic effort estimates. In contrast, starting with an estimate of the minimum effort, rather than an estimate based on ideal conditions, did not have the same positive effect on the subsequent estimate of the most likely effort.
Conclusion: The empirical results from our studies together with similar results from other domains suggest that the proposed estimation process is promising for the improvement of the realism of software development effort estimates.

1. Introduction

According to published surveys, most software projects are based on estimates that are too low [14,30,3,12,33,27,38]. These surveys typically report that the average effort overrun is about 30%. There is no convincing evidence to suggest that there has been a systematic improvement in estimation accuracy or increase in bias over time.[1] Neither are there evidence to support that the problem of inaccurate and biased estimates is removed with the use of formal estimation models instead of the use of expert judgment [1,16]. Possible reasons for the lack of benefit from formal effort estimation models in this field are that important input to the formal estimation models is judgment-based and that essential relationships are not sufficiently stable and general to enable robust estimation models [7,15]. There is, however, some evidence to suggest that there are contexts that favor the use of one estimation method over another [34,25], that some estimators are more realistic than others [17], and that there are situations in which the estimates are unbiased or even biased towards effort estimates that are too high [9]. Strategies that have been evaluated and found to reduce, but not remove, the bias towards effort estimates that are too low are the use of pessimistic scenarios [28], better use of historical data [32], and the use of estimators with highly relevant development experience [18]. Interestingly, all the above strategies for removing bias have in common that they may increase the awareness of the difference between realistic and idealistic conditions. Pessimistic scenarios may increase the awareness of what typically goes wrong in software projects. Historical data may remind the estimator of realistic scenarios for similar tasks.

E-mail address: magnej@simula.no

[1] The huge improvement in estimation accuracy from 1994 to the present day, as claimed by the Standish Group in their Chaos Reports is, as far as we can evaluate, not trustworthy. See our critique of the Chaos Report in Jørgensen, Moløkken-Østvold [20]. "How large are software cost overruns? A review of the 1994 CHAOS report." Information and Software Technology **48**(4): 297–301".

0950-5849/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2011.07.001

More development experience may make it more likely that estimator will be aware of the complexities and risks of the development work. This may be a significant finding, because people frequently have difficulty in separating idealistic from realistic assumptions when making predictions, as has been reported in numerous studies [23,13,4,28,29]. This difficulty may be an important reason for the tendency towards underestimation of software development effort. Realistic estimates of software development effort do not necessarily follow from requests to be realistic, but rather from processes that enable the estimators to better separate realistic conditions from pessimistic or idealistic ones.

This paper proposes a process for judgment-based effort estimation (expert estimation) consistent with the above findings. The process assumes that increased awareness of the difference between idealistic and realistic conditions is useful to achieve more accurate effort estimates. Although the proposed process is designed to be used in judgment-based effort estimation processes, such as work-break down estimation processes [36], the steps may potentially also be useful to ensure realistic judgment-based input to model-based effort estimation.

The remaining part of the paper is organized as follows: Section 2 describes the proposed estimation process and its motivation. Section 3 describes three studies evaluating the proposed estimation process. Section 4 discusses the results in light of other results on human judgment and exemplifies how the process may be integrated into common judgment-based software development effort estimation methods. Section 5 discusses limitations of the studies. Section 6 concludes.

2. A two-step process for judgment-based effort estimation

Well-documented cognitive and motivational mechanisms potentially contributing to idealistic assumptions in situations where the intention is to be realistic include: (i) The cognitive difficulty in separating what we want to be and what is more likely to be the outcome in terms of software project effort usage and presence of problems, i.e., "wishful thinking" [11]. (ii) The tendency to over-rate how much in control of the outcome we are, i.e., "illusion of control" [24]. (iii) The motivation to present estimates consistent with an image of ourselves as more efficient and less error prone than we really are to avoid the so-called "cognitive dissonance" [8]. (iv) The optimism-inducing effect of planning step-by-step what has to be done, i.e., the optimism caused by "looking forward" [22].

The potential presence of these mechanisms motivates the two main research questions addressed in this paper:

RQ 1: Are judgment-based software development effort estimates requested to reflect realistic conditions likely to be based on idealistic assumptions?

RQ 2: Would a process explicitly asking for effort estimates assuming ideal conditions before asking for effort estimates assuming realistic conditions improve the accuracy of judgment-based effort estimates?

The systematic tendency towards under-estimation in software development suggests a confirmatory answer on RQ 1. Furthermore, if there is an insufficient separation of ideal and realistic conditions in effort estimation, it is, as argued earlier, not unreasonable to expect that making the estimators more aware of this difference will lead to more realistic effort estimates of most likely effort, i.e., that the answer on RQ 2 will be confirmatory as well.

The process we propose and evaluate in order to answer RQ 1 and RQ 2 is a simple two-step process which we believe can easily be integrated into most judgment-based effort estimation processes. An integral part of the proposed process is the concept of "ideal effort". Ideal effort may for example be defined as the effort needed assuming that the work is completed without disturbance, full productivity all the time and no major problems. Ideal effort is in many ways similar to the concept of "ideal days" in agile estimation [6]. An essential difference to ideal days in agile estimation is, however, that we use ideal effort only as a contrast to realistic (most likely) use of effort, while the number of ideal days is the final result of the estimation process used in the planning of agile software projects. An assumed implication of our use of ideal effort as an intermediate step and not as the end result is that a consistent interpretation of "ideal" is not essential as long as the understanding of ideal effort enables the estimator to contrast what he or she considers to be the effort usage in ideal conditions with the effort usage in realistic conditions, i.e., the most likely use of effort. This way we may avoid the frequently reported challenge in agile estimation related to "my ideal days are not your ideal days" [6].

The proposed estimation process assumes that the estimator has read and understood the software requirement, preferably conducted some risk analysis and is ready to provide the estimates of the effort of the project as a whole or per activity, user story, feature, use case, etc. Instead of applying the traditional one-step process, where the estimator is requested to provide the most likely use of effort directly, we propose the use of a two-step process emphasizing the contrast between ideal and most likely use of effort:

- Step 1. Request the developer to assume that the development is completed under ideal conditions and to estimate the use of effort under these conditions. The description of the ideal condition should ensure that it is meaningful to contrast ideal with typical conditions. This implies that the described ideal conditions should deviate substantially from typical conditions, but not so much that the ideal scenario cannot be used as meaningful reference point.
- Step 2. Remind the estimator of the difference between ideal and realistic conditions and, then, request the developer to provide an estimate of the most likely use of effort (the realistic use of effort). The reminder should be sufficiently strong to trigger an estimation process contrasting ideal and typical use of effort.

3. The empirical studies

The three empirical studies described in this section compare the judgment-based estimates of most likely effort produced by the proposed two-step estimation process with those produced by the traditional one-step process. To test the robustness of the proposed estimation process, we evaluate it using different formulations of ideal conditions, different reminder formulations and different estimation tasks in the three studies. In addition, we test whether the use of the probabilistic thinking-based concept of "minimum effort", described as the effort usage only 5% likely to underrun, yields the same effect as produced with the presumably more scenario thinking-based concept of ideal effort.

The progress in results from Study A to Study C is as follows: Study A provides the first evidence in support of that the proposed estimation process lead to higher and more realistic effort estimates than the traditional estimation process. This study also reports that there is not much difference between effort estimates assuming idealistic and realistic conditions, i.e., that many developers seem to think too idealistically when estimating most likely effort. Study B replicates the results from Study A in another domain and with instructions assuming even more idealistic conditions than in Study B. Study B, in addition, reports that the use of a probabilistically defined minimum effort has not the same effect as the use of ideal effort. This supports the assumption that it is the