Estimating Software Task Effort in Crowds

Mohammed Alhamed
School of Computing Science
University of Glasgow
Glasgow, Scotland
m.alhamed.1@research.gla.ac.uk

Tim Storer
School of Computing Science
University of Glasgow
Glasgow, Scotland
timothy.storer@glasgow.ac.uk

***Abstract*—A key task during software maintenance is the refinement and elaboration of emerging software issues, such as feature implementations and bug resolution. It includes the annotation of software tasks with additional information, such as criticality, assignee and estimated cost of resolution. This paper reports on a first study to investigate the feasibility of using crowd workers supplied with limited amounts of an issue and contextual information to provide comparably accurate estimates using Planning Poker. The paper describes our adaptation of planning poker to a crowd and our initial trials. The results demonstrate the feasibility and potential efficiency of using crowds to deliver estimates. We also review the additional benefit that asking crowds for an estimate brings, in terms of further elaboration of the details of an issue. Finally, we outline our plans for a more extensive evaluation of crowd planning poker.**

***Index Terms*—Planning Poker, crowdsourcing, human computation, task effort estimation**

I. Introduction

Software maintenance consumes enormous resources, with surveys suggesting that well over half of the total cost of a software project is consumed by maintenance [10]. A key task during maintenance is the refinement and/or elaboration of new tasks, such as feature implementations and bug resolution. This work in itself is recognised as potentially a significant drain on a project's resources. For example, citing an overview of the Mozilla project, Hooimeijer and Weimer [7] states that:

"For software that is widely deployed, the number of bug reports typically outstrips the resources available to triage them."

Within this context, this paper focuses on improving the efficiency of producing effort estimations for software development tasks using the popular expert estimation processes Planning Poker [6]. Expert estimation techniques typically involve iterative structured workflows to enable the production of a consensus decision from a group of domain experts. Jørgensen [8] found that expert estimation methods generate higher accuracy estimates on average. However, such methods rely on the availability of domain experts with suitable prior experience of similar tasks to make accurate subjective judgements.

On the other hand, in crowdsourcing, large numbers of workers can be recruited to undertake tasks at relatively low cost [17]. These crowds may be used to produce a far larger number of judgements that can be orchestrated computationally. The intuition is that large numbers of workers can perform as well as small groups of experts, if the task is appropriately structured.

This paper extends the current research by investigating the application of crowdsourcing to effort estimation of software task. To do this, we report on the results of the an exploratory study to employ a crowd to estimate a series of software development tasks. Therefore the research question was: *Given a software task that required between one hour and two weeks effort, can a crowd team with an average of 24 workers produce a cost estimate that is of comparable accuracy to that of a project expert?*

Contribution: The study reported in this paper is the first study of employing crowd workers to conduct Planning Poker estimates of software development tasks. The work provides the first insights that such an organisation of crowd workers is promising and it could produce an estimate that is of comparable accuracy to that of a project expert.

This paper is structured as follows. Section II reviews related work to the present research in the existing literature. Section III presents the experimental design for investigating the efficacy of Crowd Planning Poker (CPP). The results of the experiments are then presented in Section IV. These results are discussed in Section V, together with the lessons learned from the study. Finally, Section VI summarises our conclusions from the initial study and our reflections on the next steps in the research.

II. Related Work

While we have not discovered other attempts to apply crowdsourcing to software task estimation, the application of crowdsourcing to software engineering tasks has received considerable attention in the literature, as illustrated in the recent survey by Mao et al. [14].

Similarly, there has been some work on the application of crowdsourcing to management tasks [3, 11]. Flostrand [3] provided a qualitative comparison between an in-person expert Delphi method and the use of crowds for making predictions of uncertain quantities or event outcomes.Kulkarni et al. [11] demonstrated that crowds could be used to break down complex tasks into simpler sub-tasks.

Several aspects of other experiments inspired the design of our CPP method described in the next section. In particular, Kutlu et al. [12] investigated the value of a worker's rationale in evaluating the quality of their subjective decision in a

task. The work also showed that rationales can reveal valid subjective reasons for disagreements between crowd workers on their judgements. Drapeau et al. [2] demonstrates the value of permitting negotiation between crowd workers to enhance decision making. An alternative approach to assessing quality of work is to monitor the behaviour of crowd workers [9, 4]. In these approaches, workers are expected to behave in a certain way such as seeking specific information, or spend a certain amount of time on a screen in order to have fully engaged with a task.

III. Experimental Design

In this section, we describe the the software tasks that formed the experimental objects of our study; describe our adaptation of Planning Poker for use within a crowd; our technique for filtering estimates provided by the crowd workers based on the quality of an associated justification and their behaviour; and the configuration of our study trials.

A. Baseline and Ground Truth

A set of issues were selected from an issue tracker for an open source software project, JBoss Developer Community, as the objects of the experimental study. The issue tracker was searched for issues that had been successfully resolved and been annotated with a similar estimated and actual person-time cost for issue resolution during the issue's life-cycle. The actual cost was necessary as ground truth. Taking a thorough review of issues log, actual cost is updated by the issue assignees accumulatively. Each issue may be documented with different level of details, such as issue descriptions, comments, and attachments.

The estimates and final costs are reported as literal person-hours or days in the JBoss issue tracker. However, Cohn [1] argues that software tasks are notoriously difficult to estimate accurately to a high level of precision, so teams should use relative categorical units, such as story points. The time-costs reported in the JBoss issue tracker were therefore grouped into categories so that they could be compared with approximate costs produced during a Crowd Planning Poker activity. The estimate unit labels used were one hour, half a day, one day, half a week, one week, two weeks and more than two weeks.

B. Crowd Planning Poker Workflow

Planning Poker [6] is an iterative effort estimation technique that is often associated with agile software development methods. The aim of the method is to achieve an agreed estimate amongst a team, whilst ensuring that conflicting opinions are discussed and resolved. In Planning Poker, each development team member is equipped with a set of cards, each labelled with a cost estimate.

Each team member individually estimates how much effort a software development task will take and selects the appropriate card. When all the members are ready, they then simultaneously reveal their estimates and check for estimation consistency. If the estimates are not consistent, then the team discusses the estimations before repeating the card selection and revealing process. It may take several rounds until the team reaches a consistent estimation about the task.

The general, iterative model of the Crowd Planning Poker activity to be performed by crowd workers is shown as a flowchart in Figure 1. The crowd workers are provided with the core information describing a task (a summary title and supplementary description). They may also have access to additional contextual information about the nature of the wider software project, for example, the programming language, software framework or technology platform used to develop the software and software developer comments on the the issue. In addition, workers may search for further information using a search engine query dialogue.

Once the worker has reached a decision they will be asked to submit their estimate and a short justification to complete this iteration of the task. Each estimate received was manually evaluated by the experimenter for quality according to the procedure described in Section III-C. Estimates that were determined to be invalid were removed from the experiment and not used further.

The consensus achieved between the crowd workers is calculated. If consensus is not achieved, the workers are invited to review a summary of the accepted estimates provided by the rest of the crowd, prior to submitting new estimates. The intention here is mimic the consensus forming behaviour in in-person planning poker, by allowing workers to consider the boundary estimates provided by the crowd in the previous round. For the purpose of allowing the effect of iteration on consensus forming to be studied, this process has been fixed to proceed three rounds despite the consensus score. We anticipate that if CPP was used for real estimation then the consensus calculation could be used to decide whether to terminate the activity early.

The experimental subjects were crowd workers recruited from the Amazon Mechanical Turk (mTurk) platform. Eligibility for participation in the experiment was restricted to those workers who self-declared some software development background and had been working on software development for at least two years.

C. Monitoring Quality Of Crowd Work On Subjective Tasks

We noticed during early pilots that approximately 80% of the submissions were of poor quality and that the quality of the justification appeared to be correlated with the accuracy of the estimate. To address this, we developed a multi-component task assignment quality model to determine whether a crowd worker had effectively engaged in the time-cost estimation task. The model combines a quality assessment of the workers' justification for their estimate [15]; with a scoring of their behaviour whilst on task, utilising a log trace of events recorded in the crowd task user interface [18]. The justification provided by the crowd worker was evaluated for the presence of a task breakdown, a time assignment for each working block, a general discussion about the task topic, and an explanation the estimation process applied. The behaviour of the crowd worker was developed based on the event log of

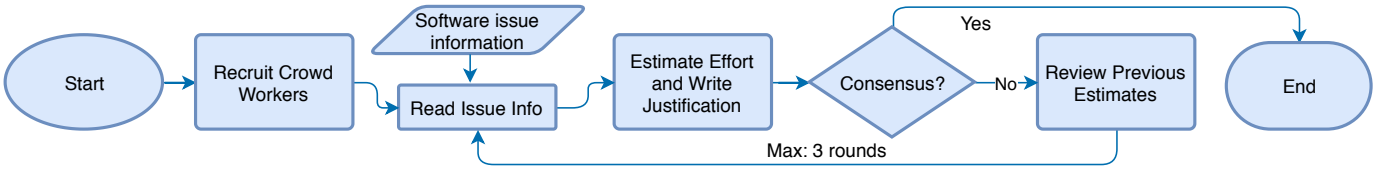

Fig. 1. General model of the crowdsourcing planning poker task

their interactions with the task assignment user interface was evaluated. Estimates were only accepted if the justification contained all the expected elements and the behaviour of the crowd worker behind these answer was required to score >75%. While the classification process has been designed to be run automatically, the scores were calculated manually as a proof of concept. Further work is needed to fully automate the scoring process.

D. Experimental Trials and Variables

Two independent variables, actual task cost (as recorded in the JBoss Issue Tracker), and task information provided to the crowd, were used for study trials. The dependent variable of the experiment is the accuracy of crowd effort estimations. Information available to the crowd worker about the estimation task were divided into two categories. Basic information comprised only issue title and supplementary description. Extended information included the ability for the crowd worker to review contextual project details, definitions of project specific terms in the issue title and description and all the comments that development team made on the given issue. In addition, crowd workers provided with the extended information were also invited to conduct a search using the Google search engine for any additional information that they required.

Five trials were conducted. Trials 1,3-5 were used to assess whether a crowd could be used to produce reliable estimates across a range of actual task sizes and address the research question. Two criteria were identified for assessing the estimates produced by the crowd planning poker process during the trials. First, it is necessary to determine if the estimates produced by the crowd are accurate when compared with those produced by the project experts. Accuracy relative to this baseline was therefore calculated as the absolute difference between the median estimate category produced by the crowd and that produced by the experts. Secondly, the extent of agreement amongst the crowd estimators is measured using Fleiss' Kappa. If the crowd is able to reach a fair level of agreement (21% to 40%) [13], then this will be considered reliable.

The impact of basic versus extended information was addressed in Trials 1 and 2. The crowd in Trial 2 were only presented basic information as compared to the extended information for the same issue in Trial 1. The outcomes from these two trials will be compared to determine the impact of the additional information on the estimates provided by the crowd, using the same calculations as above.

Separately, the extent to which crowd estimators actively seek additional information for tasks of different cost will be investigated in Trials 1, 3-5. This comparison will determine whether additional information is considered by crowd estimators when producing their estimate.

IV. Results and Evaluation

Table I summarises the results from the five trials. All five trials ran for the full three rounds, with between 13 and 37 estimates received in each of the rounds, giving a total of between 66 and 76 submitted estimates for each trial. The table also shows the estimates agreed upon by the crowd, compared with that given by the experts and the consensus in the round.

Reviewing the estimates produced by the crowds in trials 1, 3-5 it can be seen that the crowd produced an accurate estimate in three out of the four cases. The trials also suggest that the CPP process can distinguish between tasks of different cost, ranging from half a day (Trial 1) through to two weeks (Trial 5). The crowds were also able to exceed the expected threshold of consensus during the trials, reaching a consensus of at least 25% within two rounds. This suggests that the crowds do benefit from reviewing a previous round of estimates, but that the estimates then also stabilise within a few rounds.

Table I shows that for most trials rejected estimates constituted approximately 67% those submitted, although in Trial 2, this rate was higher, at 83%. Further work is needed to understand how this quality score can be automated to minimise the cost of processing judgements.

Considering the effect of information, the crowd in Trial 2, using only a basic level of information eventually agreed an estimate of Half-week, compared with the (correct) expert estimate of Half-day and achieved no consensus over three rounds. Conversely, the crowd in Trial 1, estimating the same issue with extended information produced a more accurate (if still incorrect) estimate of One Day, and did so with a fair amount of consensus by the end of the third round. Whilst preliminary, these results do suggest that a crowd benefits from additional contextual information when producing an estimate. In addition, the Extended Information Requests column shows that workers in Trials 1, 3-5 requested extended information. Further, a small subset of workers also used the provided search functionality to engage in open searches about the project. These results indicate both that the crowd workers are willing to obtain additional information in order to complete their task and benefit from doing so.

Considering costs, it was assumed that a crowd worker would require two minutes to produce an estimate. At the

TABLE I
SUMMARY OF TRIALS, INCLUDING ESTIMATES RECEIVED, OUTCOME FOR EACH ROUND AND OVERALL TRIAL AND LEVEL OF AGREEMENT ACHIEVED.

Trial	Issue	Task information	Round	Estimates Received	Considered Estimates	Crowd Estimate	Expert Estimate	Consensus (Fleiss' Kappa %)	Information Requests	Open Searches
1	1	Extended	1	16	5	Half-week		10.00		
			2	29	11	One Day		30.91		
			3	31	9	One Day		44.44		
				76	25	**One Day**	**Half-day (1)**		29	0
2	1	Basic	1	21	3	One Day		0.00		
			2	21	5	One Week		10.00		
			3	30	4	Half-week		0.00		
				72	12	**Half-week**	**Half-day (2)**		–	–
3	2	Extended	1	21	5	One Day		20.00		
			2	20	6	One Day		26.67		
			3	25	10	Half-week		20.00		
				66	21	**Half-week**	**Half-week (0)**		7	2
4	3	Extended	1	37	5	Half-week		30.00		
			2	25	13	One Day		43.59		
			3	13	7	Half-week		23.81		
				75	25	**Half-week**	**Half-week (0)**		15	5
5	4	Extended	1	32	4	Two Weeks		16.67		
			2	25	11	Two Weeks		62.22		
			3	18	9	Two Weeks		58.33		
				75	24	**Two Weeks**	**Two Weeks (0)**		47	5
Total				**364**	**107**					

time of the study, the minimum wage in the UK was £7.50, giving a cost per estimation task of 25p. The total number of estimates received was 364 , giving a total cost of £85 across all five trials, or £17 per issue. In practice, costs would have been reduced by not paying all assignees regardless of issue quality and by halting the CPP process once consensus had been achieved, resulting in a cost of £4 per issue.

Conversely, the average hourly rate of a software developer working in the UK is approximately £33 including employer costs [16]. If a small team of five developers is assumed and allowing five minutes to estimate each issue, then the cost per issue is approximately £14. This preliminary analysis suggests that crowd sourcing of software task estimates could generate a significant saving on a team's resources. Moreover, the cost compression only considered the hourly rate of workers in both environments (crowd, employment). Other costs such as deploying the crowd tasks or coordinating employees meeting have not considered. All in all, the goal is to get a general sense of the crowd task cost.

V. LIMITATIONS

Several aspects of the experimental design were limited during the study, which restricts the generalisability of the results. First, a crowd size of an average of 24 workers was selected for each round of the experiment. As can be seen from the results, this size of crowd produced the correct estimate in three out of four cases where extended information was used, and achieved 'fair' consensus [13] in all four trials, suggesting that the selected size of the crowds can produce reliable estimates. However, a shortcoming of this approach is the relatively small number of estimates given per round and within each trial in total. This problem is exacerbated by the high rejection rate experienced during the study (reaching 87% in trial 2). The study establishes a base line that the selected crowd team size can perform accurate estimations. We anticipate investigating whether this baseline can be improved upon using a larger crowd in a follow up study.

Second, we limited the number of software tasks estimated and selected the tasks manually in the study in order to limit costs. The study lacks repeated trials of software tasks with known comparable cost, but different contents. Further experimentation is needed to validate the findings of the study a wider range of software tasks.

Finally, the selection of software task issues from an open source project meant there was a risk that the crowd workers searched for the source tasks themselves before submitting their estimates. The decision was taken to use an open source repository as the use of issues drawn from a repository held internally by an organisation may have raised privacy concerns.

This risk was mitigated by encouraging workers to submit *their* estimate and not penalising incorrect estimates. The workers would also have had to register an account with the JBoss project and to convert the person time estimate to the category. There is no evidence in the logs of worker behaviour that they took these steps, although it cannot be ruled out that this was done out-of-band.

VI. CONCLUSION

In a review of Surowiecki's [19] keynote talk at Agile 2008, Grenning wrote:

> "I was wondering how Wisdom of Crowds would relate to people on agile teams doing estimation and planning. I was specifically interested in how his research applied to Planning Poker, a practice used throughout the world on agile teams" [5]

As far as we are aware, Grenning's speculation went no further than this, and the work presented in this paper represents the first study of applying crowd sourcing to Planning Poker. The paper demonstrates that a version of planning poker adapted to a crowd platform can produce estimates that are of comparable accuracy to that of a project expert, with relatively little task and contextual information. The initial results presented in this paper also illustrate how crowd workers leverage the information provided to them and shows how the combination of a rationale and their behaviour during the performance of their task can be used to filter poor quality judgements.

We are now planning a larger scale study of tasks and crowd workers to extend our results based on the lessons learned from the first study. We are also seeking to extend our analysis of these results to consider how quality assessments of crowd worker submissions can be made automatically. This work will also consider how the Crowd Planning Poker model described in this paper can be extended to mimic the discussions held within in-person Planning Poker workflows. We anticipate testing whether these discussions can produce better estimates, or help the crowd arrive at consensus more quickly.

References

[1] M. Cohn, *Agile estimating and planning*. Pearson Education, 2005.

[2] R. Drapeau, L. B. Chilton, J. Bragg, and D. S. Weld, "Microtalk: Using argumentation to improve crowdsourcing accuracy," in *Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2016, 30 October - 3 November, 2016, Austin, Texas, USA.*, A. Ghosh and M. Lease, Eds. AAAI Press, 2016, pp. 32–41.

[3] A. Flostrand, "Finding the future: Crowdsourcing versus the delphi technique," *Business Horizons*, vol. 60, no. 2, pp. 229–236, 2017.

[4] T. Goyal, T. McDonnell, M. Kutlu, T. Elsayed, and M. Lease, "Your behavior signals your reliability: Modeling crowd behavioral traces to ensure quality relevance annotations," in *Proceedings of the Sixth AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2018, Zürich, Switzerland, July 5-8, 2018.*, Y. Chen and G. Kazai, Eds. AAAI Press, 2018, pp. 41–49.

[5] J. Grenning, "Agile 2008 - wisdom of crowds keynote and planning poker," https://blog.wingman-sw.com/archives/20, August 2008.

[6] ——, "Planning poker or how to avoid analysis paralysis while release planning," *Hawthorn Woods: Renaissance Software Consulting*, vol. 3, pp. 22–23, 2002.

[7] P. Hooimeijer and W. Weimer, "Modeling bug report quality," in *22nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2007), November 5-9, 2007, Atlanta, Georgia, USA*, 2007, pp. 34–43.

[8] M. Jørgensen, "Forecasting of software development work effort: Evidence on expert judgement and formal models," *International Journal of Forecasting*, vol. 23, no. 3, pp. 449–462, 2007.

[9] G. Kazai and I. Zitouni, "Quality management in crowdsourcing using gold judges behavior," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*, P. N. Bennett, V. Josifovski, J. Neville, and F. Radlinski, Eds. ACM, 2016, pp. 267–276.

[10] J. Krogstie, A. Jahr, and D. I. S. berg, "A longitudinal study of development and maintenance in Norway: Report from the 2003 investigation," *Information and Software Technology*, vol. 48, no. 11, pp. 993–1005, 2006.

[11] A. P. Kulkarni, M. Can, and B. Hartmann, "Turkomatic: Automatic, recursive task and workflow design for mechanical turk," in *Human Computation, Papers from the 2011 AAAI Workshop, San Francisco, California, USA, August 8, 2011*, ser. AAAI Workshops, vol. WS-11-11. AAAI, 2011.

[12] M. Kutlu, T. McDonnell, Y. Barkallah, T. Elsayed, and M. Lease, "Crowd vs. expert: What can relevance judgment rationales teach us about assessor disagreement?" in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ser. SIGIR '18. New York, NY, USA: ACM, 2018, pp. 805–814.

[13] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[14] K. Mao, L. Capra, M. Harman, and Y. Jia, "A survey of the use of crowdsourcing in software engineering," *Journal of Systems and Software*, vol. 126, pp. 57–84, 2017.

[15] T. McDonnell, M. Lease, M. Kutlu, and T. Elsayed, "Why is that relevant? collecting annotator rationales for relevance judgments," in *Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2016, 30 October - 3 November, 2016, Austin, Texas, USA.*, A. Ghosh and M. Lease, Eds. AAAI Press, 2016, pp. 139–148.

[16] "Payscale," PayScale, 2019. [Online]. Available: https://www.payscale.com/

[17] A. J. Quinn and B. B. Bederson, "Human computation: a survey and taxonomy of a growing field," in *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, 2011, pp. 1403–1412.

[18] J. M. Rzeszotarski and A. Kittur, "Instrumenting the crowd: using implicit behavioral measures to predict task performance," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, October 16-19, 2011*, J. S. Pierce, M. Agrawala, and S. R. Klemmer, Eds. ACM, 2011, pp. 13–22.

[19] J. Surowiecki, *The Wisdom of Crowds: Why the Many Are Smarter Than the Few*. Abacus, March 2005.