Does the Use of Fibonacci Numbers in Planning Poker Affect Effort Estimates?

Ritesh Tamrakar[1] & Magne Jørgensen[2,3]
[1]University of Kathmandu, [2]Simula Research Laboratory, [3]University of Oslo
ritesh.tamrakar@gmail.com
magnej@simula.no

Abstract

Background: *The estimation technique Planning Poker is common in agile software development. The cards used to propose an estimate in Planning Poker do not include all numbers, but for example only the numbers 0, ½, 1, 2, 3, 5, 8, 13, 20, 40 and 100. We denote this, somewhat inaccurately, a Fibonacci scale in this paper. In spite of the widespread use of the Fibonacci scale in agile estimation, we do not know much about how this scale influences the estimation process.* ***Aim:*** *Better understanding of the effect of going from a linear scale to a Fibonacci scale in effort estimation.* ***Method****: We conducted two empirical studies. In the first study, we gave computer science students the same estimation task. Half of the students estimated the task using the Fibonacci scale and the other half a linear scale. The second study included four estimation teams, each composed of four software professionals, estimating the effort to complete the same ten tasks. Two of the teams estimated the first five tasks using the Fibonacci scale and the last five using the linear scale. The two other teams used the scales in the opposite sequence.* ***Results:*** *We found a median decrease in the effort estimates of 60% (first study) and 26% (second study) when using a Fibonacci scale instead of the traditional linear scale. The scale difference in the effort estimates decreased as the developers' skill increased.* ***Conclusion:*** *The use of a Fibonacci scale, and possibly other non-linear scales, is likely to affect the effort estimates towards lower values compared to linear scales. A possible explanation for this scale-induced effect is that people tend to be biased towards toward the middle of the provided scale, especially when the uncertainty is substantial. The middle value is likely to be perceived as lower for the Fibonacci than for the linear scale.*

Keywords*: effort estimation, Planning Poker, regression effect*

I. INTRODUCTION

Agile processes have changed not only the way many software development organizations develop their software but also the way they estimate the effort and plan their deliveries. An innovation proposed and used by many agile teams is the estimation technique Planning Poker. Planning Poker was introduced by James Grenning in a short, but highly influential, paper from 2002: "*Planning Poker or How to Avoid Analysis Paralysis While Release Planning*" [1]. Planning Poker was further developed by Mike Cohn in [2]. It may be characterized as a judgment-based, structured, group-based estimation method with elements from the Delphi method; see, for example, the Wideband Delphi method proposed by Barry Boehm [3].

When applying the Planning Poker method, a group of developers meet and go through a sequence of steps for each task, user story, or requirement to be estimated. A possible sequence leading to an effort estimate of a task is the following: i) Presentation of the task. ii) The developers discuss the task. iii) The developers produce independent estimates of the task. The estimates are typically in work-hours, ideal days, or story points. iv) The developers present their estimates, by selecting the card with the appropriate number, at the same time. v) The developers with the lowest and highest estimates justify their estimates. vi) If there is sufficient agreement on estimates, the estimation process for the task stops here. If there is a substantial disagreement or some developers have gained new insight through the other developers' justification or the discussion, then the steps are repeated from step iii). This is done until sufficient agreement is achieved or there is no change in estimates from the previous round. The group's estimate may be calculated as the mean or median of the individual estimates in the final round, or by applying other ways of consensus making. Planning Poker is believed to have several advantages compared to individual estimation and less structured group-based effort estimation, since

- Group-based effort estimation is on average more accurate than individual estimation [4].
- Independent estimation before discussing in groups is less exposed to so-called anchoring effects than unstructured group estimation [5]. Independent estimation avoids, for example, all developers being strongly influenced by the level of effort indicated by the most senior developer or the project manager early in the group discussion.

The study in [6] documented the potential benefits of Planning Poker compared to alternative combination-based estimation methods.

One central and, as far as we know, innovative element of Planning Poker, and the subject of this paper, is the non-linear sequence of numbers used for the estimation. Grenning [1] suggested that each developer should have a deck of cards with the numbers 1, 2, 3, 5, 7, 10, and "infinity," where "infinity" meant that the task was too large to be estimated, and use only these numbers for the estimation work. From his paper, it seems as if his motivation was to speed up the estimation process in an XP (eXtreme Programming) release planning context and to use a scale that reflected that the higher the estimates the less precise they are: "*It is OK to be less precise. Why invest in precision before it is needed?*" [1]. Grenning did not forbid the use of values in-between the proposed values, e.g., through adding two cards, but did warn against it: "*I bet that the added precision probably won't help a lot*" [1]. Later, possibly influenced by Mike Cohn's contributions to agile estimation [2], this has evolved into a sequence similar to the Fibonacci sequence (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …). A typical example of the sequence of numbers used by many software teams is the sequence 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100. In this paper, we will use the term Fibonacci sequence to denote sequences that are similar but not necessarily identical to the first numbers of the Fibonacci sequence.

There is empirical support for the claim that the selected Fibonacci numbers reflect the *average* level of precision of software development effort estimation. An average precision of about +/- 30% [7], for example, results in uncertainty intervals close to covering the whole scale without too much overlap or too many gaps[1].

[1] More specifically, we get for the values ½, 1, 2, 3, 5, 8, 13, 20, and 40 the corresponding uncertainty intervals [0.35; 0.65], [0.7; 1.3], [1.4; 2.6], [2.1; 3.9], [3.5; 6.5], [5.6; 10.4], [9.1; 16.9], [14; 26], and [28; 52].

In a recent, large-scale survey of empirical studies on judgment-based effort estimation in all types of domains [8], we found no empirical studies comparing the use of Fibonacci or other non-linear scales with traditional linear scales. This suggests that, in spite of the widespread use of this scale in agile estimation, we do not possess much knowledge about how this scale affects judgment-based effort estimates. More knowledge about this technique may not only improve agile estimation processes but also provide more knowledge about the mental steps of judgment-based effort estimation. In this paper, we present two studies to gain more knowledge about possible scale-induced estimation effects. The main research question is as follows:

Does the use of the Fibonacci sequence lead to different effort estimates than the use of a linear scale?

The studies are described in Section II, while the limitations, possible explanations, and implications are briefly discussed in Section III.

II. The Empirical Studies

The first study (Study 1) was conducted with university students. The main purpose of that study was to establish a context where it would be likely to find a scale-induced effect if there were any, i.e., the purpose was to demonstrate an effect rather than to examine how large it is likely to be in a field setting. For this purpose, we believe, student experiments are meaningful. The second study (Study 2) was conducted with software professionals, with realistic estimation teams and a real-world requirement specification.

A. *Study 1*

Study design: The participants were 104 computer science students following a course on software engineering at the University of Oslo, Norway. All participants received the same requirement specification, consisting of a description of a simple web-based system for registering for a summer party. The developer who had programmed this quite small and simple system had spent fewer than 8 work-hours. However, he had much more relevant experience than the typical student participant, so we would expect their effort to be higher. The students were randomly divided into two groups (Linear and Fibonacci) with exactly the same instructions, except for the scale used for the effort estimation responses. The participants were, depending on the group they belonged to, asked to put a circle around the number that best reflected what they believed was the most likely number of work-hours they would need to complete the task:

Linear Group: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, more than 40 work-hours

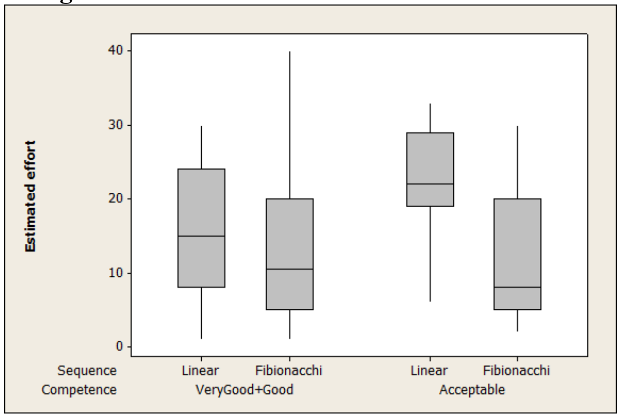
Fibonacci Group: 1, 2, 3, 5, 8, 13, 20, 30, 40, more than 40 work-hours

In addition to the estimated most likely use of effort, the developers were asked to describe their skill in solving the specified task on the scale: *very good, good, acceptable, poor*. We excluded all developers who responded with the skill category "poor" before analyzing the data, to ensure that all participants had sufficient skill for meaningful effort estimation work. After this exclusion, 89 participants remained. None of these participants used the category "more than 40 work-hours."

Results: There was a large difference in the effort estimates depending on the scale used. The median effort estimate of those in the Linear Group was 20 work-hours, while the corresponding median value for those using the Fibonacci scale was only 8 work-hours! A Kruskal-Wallis test of the difference in the median values gives $p < 0.001$. Interestingly, the median values are also the "middle" values of the two sequences, i.e., 20 work-hours is the middle value of the linear and 8 work-hours of the Fibonacci sequences. We will discuss this finding as an input to a potential explanation of the scale-induced effect in Section 3. Clearly, the effect is too large to be attributed to "rounding biases," e.g., to a tendency toward rounding the estimated effort down rather than up to the closest Fibonacci number.

The effect of the scale diminished, as can be seen in Figure 1, as the developers' skill increased. The difference in effort estimates is statistically significant, p=0.06, for the one-sided Kruskal-Wallis test of median values, even for those with the highest skill ("very good" or "good").

Figure 1: Estimates vs. Scale and Skill



B. Study 2

Design: Sixteen employees of a Nepal-based software development company conducting offshore development for a European parent company participated in the study. The company develops web, desktop, and mobile applications on ASP, .NET, and Java platforms. The participants were selected, based on their development and estimation competence, by the management of the company and paid regular fees for their work.

Following an introduction to the Planning Poker estimation method, all participants received a real-world requirement specification. The requirement specification described the web-based management of .pdf documents with scanned images. The part of the requirement specification used in the study consists of a subset of the specified requirements. In total, ten requirements were estimated by each developer and team. The estimated effort should include development and unit testing for a requirement.

Before starting the Planning Poker (group-based) estimation, all participants individually estimated the effort required to develop software that meet the ten requirements. These individual estimates were not restricted by any scale format instruction, i.e., the participants were just asked to write the number of work-hours they would need to complete the tasks. This session implies there was no difference in scale usage for the initial, individual estimate, only for teams' updates of the estimates in the Planning Poker sessions. We may therefore expect weaker scale effects than in Study 1, where the developers started with different scales.

The sixteen participants were divided into four estimation teams. Each team had four members with the same platform background (ASP, .Net, or Java) and applied the Planning Poker estimation method to their group-based estimation work. If no consensus was reached within three Planning Poker rounds, the mean value of the estimates was used as the team's final effort estimate of a requirement.

The cards used for estimating a requirement had numbers based on a Fibonacci or a linear scale:

- Fibonacci (11 cards to choose between): 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100
- Linear (100 cards to choose between): 1, 2, 3, ... 100

Two of the teams estimated Tasks 1 to 5 using the Fibonacci-scale cards and Tasks 6 to 10 using the linear-scale cards. The two other teams used the cards in the opposite sequence. As can be seen, those using the linear scale did not have the values 0 and ½ available. Since there were no estimates where it was meaningful with 0 work-hours and only one instance of a developer believing that ½ work-hour would be sufficient, we do not consider this a limitation of the comparison of the scales. Together with each individual estimate, the developers assessed their skill in completing the work on a scale from 1 (much lower than average) to 5 (much higher than average).

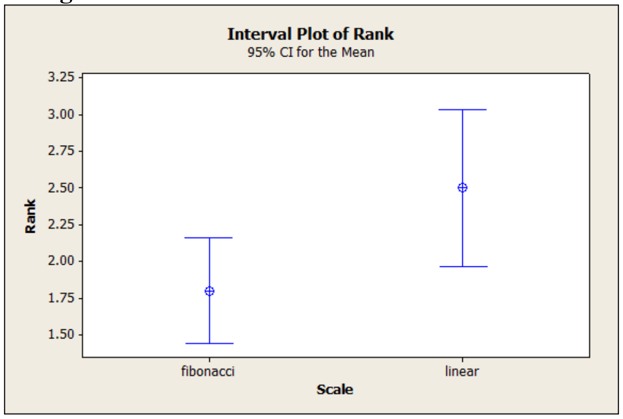
Results: There was no significant difference between the individual effort estimates of the developers starting with a Fibonacci scale and those starting with a linear scale, and the average skill level of the different teams was about the same. This suggests that any observed scale-induced difference in effort estimates is not likely to be attributed to systematic team differences in estimation optimism or skill.

The median final team estimate of a requirement was 6.75 work-hours when a team used the linear scale and 5 work-hours when the team used the Fibonacci scale. A Kruskal-Wallis, one-sided test of the difference in the median values gives p=0.07. This difference corresponds to a median 26% decrease in effort estimates when using the Fibonacci scale. The requirements were of different sizes and complexities, so this analysis is likely to be only a rough indication of the scale-induced effect.

A better analysis of the scale effects may be to rank the team estimates for each requirement, i.e., give the lowest estimate for a given requirement rank 1, the second lowest rank 2, etc. If we compare the mean rank of estimates based on the two scales, we get even stronger evidence in favor of lower estimates when using the Fibonacci scale. We find that the mean rank of the Fibonacci scale is 1.8 while that of the linear scale is 2.5; see Figure 2. An ANOVA test of the difference in mean rank gives p=0.03.

In total, we find a systematic effect from the use of the Fibonacci scale in Studies 1 and 2. This effect is especially convincing in Study 2 since all initial estimates were based on the same non-restricted scale for the individual estimates. The difference in scale usage could consequently affect only the teams' update of the estimates through the Planning Poker method.

Figure 2: Rank of Estimates vs. Scale



We separated the 40 Planning Poker team estimates into "low" or "high" skill depending on whether the team's average skill level for a particular requirement was below or above the average team skill level. We found no difference in the median values for the team estimates based on the highest level of skill (median of 5.5 work-hours for the Fibonacci and linear scale estimates). The median team estimates differed, however, significantly for those with low skill (4.75 work-hours for the Fibonacci scale and 8.0 work-hours for the linear scale). This finding further supports that the scale effects are mainly there when the estimation uncertainty is high.

III. Discussion

A. *Limitations*

The results of Studies 1 and 2 should be interpreted carefully and do not warrant strong claims about the effect of the scale in more realistic effort estimation contexts with, among other things, feedback and learning. The most important limitations, we believe, are the following:

- The estimation process applying the linear scale format, as implemented in our studies, is unusual. Our design is, we believe, nevertheless acceptable to establish that the estimation response scale matters and that we cannot expect to use non-linear scales without side effects in situations with high estimation uncertainty.
- Although the participants, especially in the second experiment, had previous estimation experience, they had no or little experience with the applied estimation formats. Consequently, it is possible that the effects will be weaker or even removed for teams more experienced in the use of, for example, Planning Poker.
- The tasks estimated were quite small, although typical for agile estimation contexts. However, we do not know much how the use of non-linear scale affects the estimation of larger tasks and projects.
- We do not know how much effort the developers would actually use. The frequently reported tendency towards under-estimation of software development effort suggests that the use of the Fibonacci scale would contribute to increased estimation error, but we need studies with knowledge about the actual effort to provide strong evidence for this.

The above limitations imply a need for further studies to better understand how the use of non-linear scales affects the effort estimates. Our studies should be considered only as a first step toward better understanding of a phenomenon that is not only theoretically interesting but also may have practical consequences for improved estimation accuracy in agile and other contexts.

B. A Possible Explanation of the Effect

The perhaps most striking difference between the formats of the Fibonacci scale and the linear scale is the difference in "gravity," i.e., that the Fibonacci sequence has a much larger proportion of its values at the lower end of the scale compared to the linear scale. Assume that we, for example, stop the sequences at 40. The linear scale would then have "20," while the Fibonacci sequence would have "8" as the middle number. The middle, or "central" value, is well known to be influential in quantitative estimation. Hollingworth wrote the following in 1910 [9]: "*Judgment of time, weight, force, brightness, extent of movement, length, area, size of angles, have all shown the same tendency to gravitate toward a mean magnitude*[.]" This tendency is denoted "the central tendency of judgment" and seems to be a robust phenomenon. Explanations of the central tendency of judgment include "anchoring" and "Bayesian updating-processes" [10]. An anchoring effect may occur, for example, when the central value is the starting hypothesis for the estimation and there is an insufficient adjustment up or down from this value [11]. The use of a Bayesian judgment-process implies that people weight the average (the prior information about the phenomenon) and the particular signal. If the knowledge is good (the signal is reliable), then the average is weighted less, but if the knowledge is poor, the average is weighted more. When the skill level increases, there is less reason for the middle value of the scale to act as the central value that uncertain estimates should regress toward. The finding that increased skill led to less effect from the use of the Fibonacci scale, therefore, seems to be consistent with the above explanation.

C. Implications

Our studies suggest that the choice of scale may have an effect on the effort estimates, especially when the estimation uncertainty is high. This, in turn, suggests that software developers should be careful when applying non-linear scales, such as the Fibonacci scale. Although using a Fibonacci scale may speed up the estimation process and reflect the precision of the effort estimates better than linear scales, a Fibonacci scale may also bias the estimates toward too low effort values. Especially in situations when there is a tendency toward over-optimism and the estimation uncertainty is high, it may be risky to use the Fibonacci scale in effort estimation. More knowledge about the tasks to be estimated may reduce the effect, and thus, the first iterations or releases might have the highest risk of under-estimation due to the use of the Fibonacci scale.

If the central tendency of judgment is a valid explanation for the finding, we hypothesize that similar findings will be observed for other types of non-linear scales. The exponential sequence, which is sometimes recommended in agile estimation, for example, might lead to an even stronger tendency toward lower values than the Fibonacci scale.

REFERENCES:

1. Grenning, J., *Planning Poker*, 2002, Renaissance Software (/renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf).
2. Cohn, M., *Agile estimation* 2006, Englewood Cliffs, New Jersey: Prentice Hall.
3. Boehm, B.W., *Software engineering economics* 1981, Englewood Cliffs, New Jersey: Prentice-Hall. XXVII, 767 s.
4. Moløkken-Østvold, K. and M. Jørgensen, *Group processes in software effort estimation*. Empirical Software Engineering, 2004. **9**(4): p. 315-334.
5. Aranda, J. and S. Easterbrook, *Anchoring and adjustment in software estimation*. Software Engineering Notes, 2005. **30**(5): p. 346-355.
6. Moløkken-Østvold, K., N.C. Haugen, and H.C. Benestad, *Using planning poker for combining expert estimates in software projects*. Journal of Systems and Software, 2008: p. 2106-2117.
7. Jørgensen, M., *A review of studies on expert estimation of software development effort*. Journal of Systems and Software, 2004. **70**(1-2): p. 37-60.
8. Halkjelsvik, T. and M. Jørgensen, *From origami to software development: A review of studies on judgment-based predictions of performance time*. To appear in Psychological Bulletin, 2012.
9. Hollingworth, H.L., *The central tendency of judgment*. Journal of Philosophy, Psychology and Scientific Method, 1910. **7**(17): p. 461-469.
10. Grieco, D. and R.M. Hogarth, *Overconfidence in absolute and relative performance: the regression hypothesis and Bayesian updating*. Journal of Economic Psychology, 2009. **30**: p. 756-771.
11. Kahneman, D., P. Slovic, and A. Tversky, *Judgment under uncertainty: Heuristics and biases* 1982, Cambridge, United Kingdom: Cambridge University Press.