Title page

Title: Relative Estimation of Software Development Effort:
It Matters With What and How You Compare

Author: Magne Jørgensen
Simula Research Laboratory & University of Oslo magnej@simula.no
Phone: +47 924 333 55

Relative Estimation of Software Development Effort: It Matters With What and How You Compare

Magne Jørgensen

Abstract: Software development effort estimation is frequently based on comparing the effort of one task relative to that of another. We present empirical results that show how relative estimation may result in biased assessments of similarity and over-optimistic effort estimates. We observe that tasks tend to be assessed as more similar than they in reality are when compared with each other, that the similarity of two tasks depends on the direction of the comparison and that it matters whether the comparison is based on difference in work-hours or as the ratio. We use the observations and other evidence to suggest ways of improving the accuracy of relative estimation.

Keywords: D.2.9.b Cost estimation, software psychology, relative estimation, story points

1. Introduction

Most human judgment includes conscious or unconscious comparisons of a target object with one or more reference objects. The comparisons are known to sometimes lead to judgment biases. We perceive the moon as larger the closer it gets to the horizon. We feel water at room temperature as warmer, if the hand just before has been submerged in cold water. We estimate the weight of a giraffe to be lower, when first asked about the weight of and comparing it with the much smaller raccoon [1].

In this article we present the results of a family of empirical studies that show comparison-related judgment biases in the estimation of software development effort. We argue that higher awareness of these biases are essential for the improvement of software effort estimation practice, particularly for judgment-based effort estimation (expert estimation), which is the approach most used in the software industry [2]. Our suggestions for improved estimation practice are based on the results from the study and other knowledge about comparison biases.

2. The Empirical Studies

If there were no estimation bias from the choice of reference task and System A was clearly much larger than System B, we would expect that the estimation request: "*How much more effort would you need to develop System A compared to System B?*" would lead to the same responses as the request "*How much less effort would you need to develop System B* *compared to System A*?" Similarly, if one group of software developers estimated System A to be about four times larger than System B, we would expect a similar group of developers would estimated System B to be about one quarter of System A. Not least, we would expect that if one group of developers estimated a system to require less effort of another system, a similar group of developers would estimate the other system to be the larger that the first system. As we will see, the actual observations of software professionals' estimates did not support any of these expectations.

The participants of our studies worked as software developers and managers in software companies in Ukraine, Vietnam and Thailand. They were required to have at least six months experience with development of web-solutions, practice with the estimation of the effort of own work and good English skills. The companies they worked in were compensated for their developers' work. In most cases, this was done through ordinary payment per hour for the estimation work.

The developers completed estimation work based on the reading of various real-world software development requirement specifications. The time available for each task estimate was 20-30 minutes. This enabled sufficient time for rough estimates of the tasks, which were not very large, but hardly enough to conduct thorough estimation work based on a detailed break-down of activities. A previous study suggests that situations with relatively high estimation time pressure and smaller tasks, such as in our studies, sometimes increases estimation biases compared to typical field settings but are not likely to create biases not there [3]. We therefore assessed the design of the study to be sufficiently realistic to provide results transferable to situations with more thorough estimation work and larger tasks. The time pressure in the studied context does, however in particular resemble real-world contexts where there is limited time for effort estimation, for example when a client requires a quick ball-park estimate of the cost of a new feature.

When a developer participated in more than one of our studies (never more than two studies per participant), the sequence and the treatments of the studies were independent of each other to avoid carry-over effects.

Through asking the developers in charge of the original development, we knew the approximate actual effort of some of the systems included in the studies. We use this effort to indicate whether the responses, on average, are likely to be realistic or not and to assess the degree of assimilation, where assimilation is understood as the degree the developers perceived two tasks to be more similar than they in reality were.

Study 1: Estimated difference in work-hours

Sixty-nine software developers received the same two requirement specifications. The first specification (System A) described the development of a web-based system for storage and retrieval of information about scientific studies. (We send the requirement specifications used in this and the other studies on request to interested readers.) The actual effort of System A was more than 200 work-hours. The second specification (System

B) described a much smaller and simpler web-based site for a summer party participation registration. The actual effort of System B was less than 20 work-hours. The actual difference in work-hours between the two systems was consequently at least 180 work-hours.

The developers were randomly divided into two groups. Those in the first group (Group Ref-A) were requested to use the larger System A as their reference and respond on the format: "*I think I would need about _____ less work-hours to develop and test System B compared to System A.*" Those in the second group (Group Ref-B) were requested to use the smaller System B as their reference and respond on the format: "*I think I would need about _____ more work-hours to develop and test System A compared to System B.*" Following the estimate of the difference between the two tasks, the participants were requested to estimate the work-effort they would need to develop System A.

The results are displayed in Table 1. We use the median instead of the mean effort estimates in this and the following analyses to avoid that a few very high estimates dominate the average values.

Table 1: Results from Study 1

	Estimate of difference	Estimate of System A
Group Ref-A	40 work-hours	64 work-hours
Group Ref-B	80 work-hours	150 work-hours

As can be seen in Table 1, those in Group Ref-A estimated that the Systems A and B were more similar in terms of effort usage (40 work-hours) than those in Group Ref-B (80 work-hours). A one-sided Kruskal-Wallis test of difference in median values gives p=0.005. While this estimation asymmetry may appear counter-intuitive, it is in accordance with Tversky's well-documented theory of feature matching, see Box 1.

Box 1: Tversky's theory of feature matching

Amos Tversky proposed in 1977 that a comparison between two objects are based on matching features of the target object with those of the reference [4]. The important part of his theory, in the context of effort estimation, is that this matching process leads to a neglect of features only present in the reference. When the objects are different, the amount of neglected features depends on the direction of the comparison and lead to asymmetries in similarity. Tversky found, for example, that people assessed North Korea to be more similar to China, than China was to North Korea. China has more unique features than North Korea. Comparing North Korea with China will therefore lead to neglect of more unique features than comparing China with North Korea. Similar direction-of-comparison asymmetries have subsequently been observed in many types of comparisons and domains. Comparing traffic with industry or industry with traffic led for example to different opinions of whether traffic or industry were most to blame for air pollution [5]. There may consequently be substantial power in controlling the direction of a comparison.

The actual difference in development effort of Systems A and B is much higher than the difference estimated by the developers, which means that we in this case had a quyite strong assimilation effect. While there are contexts where tasks get less similar (a contrast effect) when we compare them, the body of evidence from other domains suggests that contrast effects are rare and that assimilation effects are by far more common [6].

Those in Group Ref-A did not only believe that the two systems were more similar than those in Group Ref-B, but they were also substantially more optimistic about the required effort to develop System A. Recalling that the actual effort of System A was at least 200 work-hours, a median estimate of only 64 work-hours is likely to be much too low. The use of System A as the reference task did, consequently, not only affect the assessment of similarity, but also lead to too low estimates of work-effort. A Kruskal-Wallis one-sided test of the difference in the groups' median estimates of System A gives p=0.005. The estimate of System A subtracted the estimate of the difference between Task A and B, see Table 1, suggest that those in Group Ref-B would have over-estimate the effort of Task B. This is also likely to be an effect of the strong assimilation effect observed.

Study 2: Estimation Sequence

The direction-of-comparison effect found in Study 1 may not be restricted to explicit comparisons of the effort of two tasks. When estimating the effort of two related systems in a sequence, it is likely that the first is included in the references for the second. In accordance with Tversky's feature matching theory and the results of Study 1, we would expect that the estimated effort of a system increases compared to another when that system is used as the target compared to when it is used as the reference.

This expectation can be tested in a context where two Systems C and D require about the same work-effort and are estimated in a sequence. Software developers first estimating System C and then System D would then tend to believe that System D is the larger of the two, because System C is used as reference. Software developers first estimating System D and then System C would, on the other hand, use System D as target and believe that System C is the larger. This is an effect similar to the one reported in [5], when the main contributor to air pollution out of traffic and industry depended on the direction-of-the-comparison.

We requested 35 software developers to estimate the effort of two systems of similar size. One of the systems was a web-based system for membership management of a large software development organization (System C) and the second a system for the management of doctors' appointments (System D). We had had several companies estimate the effort of the systems earlier. Their estimates, on average around 300 work-hours for each of the systems, suggested that the two systems typically were perceived to require about the same level of development effort.

The developers were randomly divided into two groups. Those in Group Ref-C first estimated the effort they would need to develop and test System C and then System D, while those in Group Ref-D estimated the systems in the opposite sequence. The median values are displayed in Table 2. (Notice that the median difference between two systems, for the same developer, will not necessarily be the same as the median estimate of one system subtracted the median of the other.)

Table 2: Results from Study 2

Group	Estimate of System C	Estimate of System D	Estimate D – Estimate C
Ref-C	220 work-hours	275 work-hours	66 work-hours
Ref-D	320 work-hours	240 work-hours	-70 work-hours

As expected, those in Group Ref-C tended to believe that System D was the larger, while those in Group Ref-D tended to believe the opposite. A one-sided Kruskal-Wallis test of difference in median values of Estimate D – Estimate C of the two groups gives p=0.01. When looking at the groups' first estimates, System C for those in Group Ref-C and System D for those in Group Ref-D, we see that the developers – when not impacted from the sequence - would have estimated the two systems to require about the same level of effort.

The results from Study 2 demonstrate that effects similar to those reported in Study 1 do not require explicit comparisons. They may also be present when the comparisons are more implicit, for example induced by an estimation sequence.

Study 3: Relative estimates as ratios

Sometimes the comparison between the effort of two systems A and B is ratio-based. This happens for example when we assess one system to require twice as much (200% of) or one quarter of (25% of) the effort of another system. Study 3 aims at examining whether the asymmetries observed for differences in work-hours are present for ratio-based relative estimation, as well.

We used the same two specifications (System C and D) as in Study 2. There were 34 developers, different from those in Study 2, participating. The developers were randomly divided into two groups. Group Ref-C participants were requested to respond on the format: *"I think that the work-hours I need to develop and test System D is about ______% of the work-hours I would need to develop System C."* Group Ref-D participants were requested to respond on the format: *"I think that the work-hours I need to develop and test System C is about ______% of the work-hours I would need to develop System D."* If the same effects were present on the ratio-based format as for the work-hours based format in Study 2, we would expect estimates that reflected that the target system required most effort, i.e., average responses higher than 100% for both groups.

What we found, however, was the opposite! The median response of those in Group Ref-C was that target System D was 78% of the reference System C, and, of those in Group Ref-D that the target System C was 70% of the reference System D. A Kruskal-Wallis one-sided test of the difference in responses gives p=0.07. If we transform the estimates of work-hours in Study 2 to the ratio formats of Study 3, we clearly see the difference in effect dependent on the request format. In Study 2 those using System C as the reference believed that System D was 125% of System C and those using System D as the reference that System C was 130% of System D. The difference to the corresponding 78% and 70% in Study 3 is large.

One possible reason for the result in Study 3 is that people tend to expect that percentage-based requests are formulated so that it is possible to respond with values less than 100%. In other words, the software developers receiving the request "how many percentage of X is Y?" may assess that the person asking expects X to be smaller than Y and get biased by this expectation. Several studies suggest that there can be a strong biasing effect from "reading between the lines", that is from inducing what the person requesting the judgment think, even with highly experienced developers [5]. We report in [7] that describing a task as a "minor change" compared to as "new functionality" leads to lower estimates, even with highly experienced developers. Regardless of the validity of the above explanation, the results of Study 3 demonstrate that we cannot expect that the outcome of comparison processes are independent on whether we ask for comparisons based on differences in work-hours or as ratios.

Study 4: Story point-based estimates

Story point-based estimation is frequently explained through "times larger" or "percentage of". We therefore hypothesized that story point-based estimation would have estimation asymmetries similar to what we found in Study 3. To test this we trained 62 developers, different from those in the Studies 1-3, in the use of story point-based estimation through a set of instruction material, examples and training tasks. We then divided the developers randomly into two groups. Those in Group Ref-SMALL were instructed to use a small user story as their reference user story (their baseline user story) and estimate a substantially larger user story. Those in Group Ref-LARGE were instructed to use the larger user story as the reference and estimate the smaller user story. In both groups the reference user story was given 10 story points.

We found that story point-based estimation, as expected, had the same asymmetries as for the "percentage of"-requests in Study 3. The median number of story points of Group Ref-SMALL was 23 (the larger is 230% of the size of the smaller), while the median respons of those in Group Ref-LARGE was 3 (the smaller is 3/10 = 30% of the size of the larger, which rationally speaking correspond to the belief that the larger is 10/3 = 333% of the size of the smaller). A Kruskal-Wallis one-sided test of difference in corresponding median values gives p=0.06. The actual effort of the user story, as provided by the original developer, suggest that the largest user story required at least 3 times more effort than the smaller one, i.e., that those in Group Ref-SMALL underestimated the real difference.

Supporting studies

To further test the robustness of the above findings we asked the software developers to conduct relative estimation of the population of various countries. The results of these additional studies support the generality of our findings for quantitative estimation. The median responses of a selection of the additional studies are:

- USA has 155 million more inhabitants than Mexico (Mexico is the reference), but Mexico has only 100 million inhabitants less than USA (USA is the reference). This result is similar to that of Study 1.
- Poland has 10 million more inhabitants than Romania, when first estimating Romania and then Poland (Romania is the reference), but about the same number of inhabitants when first estimating Poland and then Romania (Poland is the reference). This result is similar to that of Study 2.
- Austria is 70% of Hungary (Hungary is the reference), and Hungary is 80% of Austria (Austria is the reference). This result is similar result to that of Study 3.

All above differences were statistically significant at p<0.05 using Kruskal-Wallis test of difference in median values.

Threats to Validity

Earlier, we argued that the realism of the studies is sufficient to expect the results to be valid in field settings. One counter-argument could be that the developers did know that they would not have to complete the tasks and were consequently less motivated for producing accurate effort estimates. To test the effect of increased motivation for accuracy, we completed a follow-up study with design similar to that in Study 1, but with monetary incentives for estimation accuracy. Five among the 50% most accurate estimates would win an Amazon gift card worth 100 USD. Forty-five computer science students participated. The study gave results very similar to those in Study 1. An additional argument for the external validity of our findings is that the results in Study 1 are consistent with the bidding results of the field study documented in [8]. In that study, the companies who used a larger system as reference for a smaller one, on average assessed the difference in effort between two systems to be the less than those using the smaller system as reference for the larger.

It is, nevertheless, likely that the biases we observed in Studies 1-4 will not be strong or even present in several real-world context. One condition, in particular, that may reduce the biases is the presence of extensive knowledge or experience. We found, for example, no biases when we asked developers from Ukraine about something where they had much knowledge, that is the difference between the populations in Russia and Ukraine. Similarly, it is likely that software professionals being asked about the relative effort of two tasks will be less or not at all exposed the observed effects if they have extensive knowledge about the compared tasks.

Practical implications for estimation practice

The studies imply that the choice of reference and the format of comparison matters and can lead to systematic software development effort estimation biases. The following suggestions for estimation practice are partly based on the findings in Studies 1-4 and partly other relevant results:

1. When using analogy-based estimation on project level, use reference projects similar to the project to be estimated and compare the projects through difference in work-hours rather than "percentage of".

Evidence: While not likely to remove all comparison biases, it is in light of the observed strong assimilation effect (Studies 1 and 2) and the risk connected with using "percentage of"-based estimation of similar projects (Study 3), better to choose similar than much smaller or larger reference tasks and base the comparison on difference in work-hours.
Comment: We have previously observed a reluctance to use dissimilar projects as analogies, and adjust for the differences [9]. While at that time surprised by this reluctance, which had the unfortunate consequence that the estimation teams had problems applying analogy-based estimation methods properly due to lack of identified analogies, we now see that there are good reasons for this reluctance.

2. Be aware of and try to compensate for the tendency towards neglect of properties unique for the reference task or project.

Evidence: See Tversky's feature matching theory in Box 1, which is consistent with our findings in Studies 1 and 2.

3. When estimating the effort in work-hours of a sequence of tasks, start with the estimation of a medium large task and avoid sequences that lead to large difference between one task and the following.

Evidence: The knowledge about the previously estimated task is typically highly accessible and that task is therefore likely to be used as reference, see Study 2. See also our argumentation in 1 on the importance of similarly sized analogies.
Comment: There are many concerns when estimating tasks in a sequence, for example to estimate logically connected tasks together. If not feasible to change the estimation sequence, the awareness of possible sequence biases may reduce the comparison bias.

4. When using ratio-based estimation, such as story points, avoid the use of a small user story (task) as reference.

Evidence: The use of a small user story as reference in Study 4 made the user stories too similar (assimilation effect) to reflect the real difference.
Comment: A study (unpublished) we conducted with twenty-one software professionals in a Polish outsourcing company replicated the finding in Study 4, and suggested that the use of median large user story led to least assimilation bias.

5. Avoid requests for comparisons on formats where it is likely that the estimator will start "reading between the lines". In particular, this includes the request "how many % of the effort required to develop X is required for Y?" in situations with similarly sized projects.

Evidence: Software developers are easily affected, sometimes unconsciously, by what they think the person requesting an estimate believes or desire about the outcome. See Study 3.

6. Combine relative estimates derived from independent processes using different request formats.

Evidence: As shown in the Studies 1-4 estimation biases may reverse dependent on the request format. Numerous studies suggest that independent combination of estimates based on independent and different approaches will give more accurate judgments [10].

References:

1. Frederick, S.W. and D. Mochon, *A scale distortion theory of anchoring.* Journal of Experimental psychology: general, 2011. **to appear**.
2. Hihn, J. and H. Habib-Agahi. *Cost estimation of software intensive projects: A survey of current practices*. in *International Conference on Software Engineering*. 1991. Austin, TX , USA: IEEE Comput. Soc. Press, Los Alamitos, CA, USA.
3. Jørgensen, M. and S. Grimstad, *The Impact of Irrelevant and Misleading Information on Software Development Effort Estimates: A Randomized Controlled Field Experimen.* IEEE Transactions of Software Engineering, 2011. **To appear**.
4. Tversky, A., *Features of similarity.* Psychological Review, 1977. **84**(4): p. 327-352.
5. Wänke, M. and L. Reutner, *Direction-of-comparison effects: How and why comparing apples with organges is different from comparaing organges with apples*, in *Perspectives on framing*, G. Keren, Editor 2010, Psychology Press (Taylor and Francis Group): New York and Hove.
6. Mussweiler, T., *Comparison processes in social judgment: mechanisms and consequences.* Psychological Review, 2003. **110**(3): p. 472-489.
7. Jørgensen, M. and S. Grimstad, *Avoiding irrelevant and misleading information when estimating development effort.* IEEE Software, 2008. **25**(3): p. 78-83.
8. Jørgensen, M., *The effects of the format of software project bidding processes.* International Journal of Project Management, 2006. **24**(6): p. 522-528.

9. Jørgensen, M., *Top-down and bottom-up expert estimation of software development effort.* Information and Software Technology, 2004. **46**(1): p. 3-16.
10. Armstrong, J.S., *Combining forecasts*, in *Principles of forecasting: A handbook for researchers and practitioners*, J.S. Armstrong, Editor 2001, Kluwer Academic Publishers: Boston. p. 417-440.

Biography: Magne Jørgensen works as a researcher at Simula Research Laboratory and professor at University of Oslo. Previously, he worked with software development, estimation and process improvement in the telecom and insurance industry. He has, together with prof. Kitchenham and dr. Dybå, founded and promoted evidence-based software engineering and teaches this to both students and software professionals. His current main research interest is judgment-based effort estimation, in particular in the context of very large software projects.