Cover page:

The Impact of Customer Expectation on Software Development Effort Estimates

Prof. Magne Jørgensen*

Simula Research Laboratory

P.O.Box 134, NO-1325 Lysaker, Norway

+ 47 924 333 55 (Telephone), +47 67 82 82 01 (fax)

magne.jorgensen@simula.no

Prof. Dag I. K. Sjøberg

Simula Research Laboratory

P.O.Box 134, NO-1325 Lysaker, Norway

dagsj@simula.no

* Corresponding author.

The Impact of Customer Expectation on Software Development Effort Estimates

ABSTRACT

The results from the study described in this paper suggest that customer expectations of a project's total cost can have a very large impact on human judgment-based estimates (expert estimates) of the most likely use of software development effort. The information that the customer expectations are not valid estimation information did not remove the impact. Surprisingly, the estimators did not notice this impact or assessed it to be low. An implication of the results is that the provision of realistic project estimate of most likely use of effort may require that the estimators do not know the customer's expectations of the total cost of the project.

Keywords: Managing projects, project effort estimation, project planning.

1. INTRODUCTION

Organisations developing software have, in general, a bad reputation for effort overrun. According to a survey carried out by Standish Group, see http://www.standishgroup.com, the average software project cost overrun was as high as 189% of the original estimate, and only 17% of the projects completed on-time, on-budget and with all features and functions as initially specified. Large effort overruns may lead to dissatisfied customers, low quality of the software and frustrated software developers.

This paper focuses on “expert estimation” of software development effort, i.e., estimation conducted by an person, who is recognized as an expert on the task, following a process that is, to some degree, non-explicit and non-recoverable. According to several empirical studies, expert estimation is the most common software project effort estimation approach [1-4]. Hence, it is important to understand why expert estimates are biased toward too low values. Possible causes for, on average, too low expert estimates of software development effort are, for example:

- an ‘inherent’ over-optimism when estimating effort in situations with high uncertainty,
- a lack of separation of bid (price-to-win) and realistic effort usage when competing with other development organisations to get a project, and,
- in cases with too high effort estimates, the remaining effort is used to improve the delivered product (Parkinson’s law).

We believe that all these three causes are important. The study described in [5] reports that the treatment of uncertainty in the evaluation of projects does not take into account the imbalance of negative over positive unknown events. Frequently, the likelihood of negative events is higher than the likelihood of positive events, while people’s estimates are typically based on equal likelihood. In [6] it is reported that software development organisations often blur the distinction between price-to-win and realistic estimate. The resulting estimate is then becoming a mixture of both. In a bidding situation this means that there will be a tendency towards too low estimates of most likely effort. Project leaders we have interviewed state that they seldom use less effort than planned, because additional budget time is then used to improve the product. This means that much of the apparent bias of the estimate is caused by a different project behaviour when the estimates are too low compared with when they are too high. Similar results are described in [6-9].

In this paper we focus on a fourth possible cause of effort overruns:

- an impact of the customer's expectations regarding effort usage.

The research focus on this cause was motivated by the observation that much too low effort estimates frequently were connected with customer expectations of very low cost [6]. A typical scenario was that a development project had communicated to the customer an early, much too optimistic effort estimate based on very limited knowledge. When the project leader initiated the ordinary estimation process, it was known that the customer expected an amount of effort not far from this early estimate. The ordinary estimation process led to an effort estimate significantly higher than the early estimate, but still much too optimistic. In a student experiment [6], it was found that effort estimates could easily be impacted by irrelevant information, even when the students knew that the information was irrelevant. These two findings together suggested that even in situations where the development organisation knows that the customer expectation about effort usage is unrealistic, the estimates may get impacted.

We formulated the following two research questions (RQ):

RQ1*: Are software development effort estimates impacted by irrelevant information about the customer expectations?*

RQ2*: To which degree are software developers aware of the (potential) impact of irrelevant information about the customer expectations?*

Although potentially important to understand the frequency of over-optimistic effort estimates we have been unable to find any previous study on these research questions.

The remaining sections of this paper are organised as follows: Section 2 describes and discusses the design of the experiment. Section 3 reports the results. Section 4 describes and analyses possible reasons for the findings. Section 5 briefly outlines implications of the findings for effort estimation work. Section 6 discusses threats to validity of the reported results. Section 7 concludes the study.

2. DESIGN OF THE EXPERIMENT

It is difficult to isolate the impact of the customer expectation from other types of estimation impacts when observing or participating in real-life estimation and development work [10]. For this reason, we decided to conduct a controlled experiment using computer science students and software professionals instead of applying the case study research method. One advantage of including both students and software professionals is that we can then study the usefulness of students as subjects in this type of software estimation studies.

2.1 Subjects

At the Department of Informatics of the University of Oslo, 38 computer science students were paid to participate in the experiment. The students had, on average, programmed 6000 lines of code, completed 6 courses in informatics and worked 1.5 months as a programmer in the software industry. All the students had programmed at least 1000 lines of code. The 12 professional software developers belonged to the same company and developed e-commerce applications. The company was paid for our use of their software developers. The software developers had on average 4.6 years of experience. All of them had experience in estimating the effort of own work, and five of them had been in charge of total project estimates in the role as project leader.

2.2 Estimation Task

The participants were asked to estimate how much effort they would need to develop a specified software system and to describe their estimation strategy. The requirement specification described a simple, real-life, administrative system for time shift swapping at a hospital. The estimators were free to select the development technology they knew best to ensure that they had sufficient experience to enable meaningful estimation of effort. The estimators should follow a bottom-up estimation strategy, i.e., they were requested to provide a list of activities necessary to complete the development and an effort estimate of each activity. The total effort was then the sum of the effort of the activities.

2.3 Treatment

The students and software professionals were divided into three groups with approximately equal size: The control group (CONTROL), the high customer expectation group (HIGH), and the low customer expectation group (LOW). The CONTROL group received no explicit customer expectation information, while customer expectation in the HIGH and LOW groups were introduced in the respective descriptions of the task, as follows:

HIGH group: *"The customer has indicated that he believes that 1000 work-hours (corresponds to about 80 000 $ in development costs) is a reasonable effort estimate for the specified system. However, the customer knows very little about the implications of his specification on the development effort and you shall not let the customer's expectations impact your estimate. Your task is to provide a realistic effort estimate of a system that meets the requirements specification and has a sufficient quality."*

LOW group: *"The customer has indicated that he believes that 50 work-hours (corresponds to about 4 000 $ in development costs) is a reasonable effort estimate for the specified system. However, the customer knows very little about the implications of his specification on the development effort and you shall not let the customer's expectations impact your estimate. Your task is to provide a realistic effort estimate of a system that meets the requirements specification and has a sufficient quality."*

The high and low expectation values were based on the actual effort used by computer science students when developing an almost identical system as part of a course in software engineering. The median effort used then was about 250 work-hours, ranging from less than 100 to more than 300 work-hours. In other words, while 50 work-hours would be very low, 1000 work-hours would be a very high effort usage for the students. We had no project data indicating what would be a very low and high effort usage for the software professionals.

After completing the estimates, the HIGH and LOW groups were asked about the impact of the customer expectation:

"Do you believe that the customer expectations had an impact on your estimate?"

Participants who answered yes on this question were asked to indicate how large they believed this impact was:

"Assess the size of this impact."

2.4 Measures and Hypotheses

We expected a rather large variation in effort estimates and could not exclude very low or very high effort estimates. Hence, several of our measures are based on the more robust median estimate instead of the mean estimate.

Based on the median values we hypothesised that:

H1: *The median effort estimates of the HIGH group (MedEffHigh) is higher than the median effort estimates of the LOW group (MedEffLow).*

H2: *The median effort estimate of the HIGH group (MedEffHigh) is higher than the median effort estimate of the CONTROL group (MedEffControl).*

H3: *The median effort estimate of the CONTROL group (MedEffControl) is higher than the median effort estimate of the LOW group (MedEffLow).*

These hypotheses are related to research question 1 (RQ1) and are motivated by the observations of customer expectations and effort estimates described in Section 1. We decided to test the hypotheses using the non-parametric rank based Kruskal-Wallis test. A non-parametric test was chosen because we could not assume that the estimated effort followed a normal or another parametric distribution.

Our hypotheses regarding research question 2 (RQ2) were:

H4*: The awareness of the (potential) impact of the customer expectation on the effort estimate is low.*

H5*: The (potential) impact of the customer expectation on the effort estimate is not correlated with the awareness.*

H4 and H5 are motivated by findings based on a number of interviews with project leaders and software developers described in [6]. The interviewed subjects did not believe that the customers' expectations had large impacts on their estimates. H4 is tested rather informally since we are unable to decide in advance what we would classify as 'low awareness'. This depends on both the actual impact, the proportion of participants admitting and impact and the self-assessed size of the impact. We test H5 using the Kruskal-Wallis test on the difference of median estimates.

It is difficult to provide arguments for a specific pre-determined significance level for tests of the hypotheses. Therefore, as suggested in [11], we present the actual p-values for the test and use this p-value, together with other relevant information in our argumentation for or against rejection of the null hypotheses (the negations of H1, H2, H3 and H5). An important part of this argumentation is the relationship between the required p-level and the number of tested hypotheses, see [12] for an introduction to this topic.

The p-value of hypothesis testing only indicates the *presence* of a difference, not the size of the difference. To measure the size of the difference in mean values between two groups, we use Cohen's size of effect measure d [13]. Conventionally, a d of about 0.2 indicates a small size of effect, a d of about 0.5 indicates a medium size of effect, and a d of about 0.8 or more indicates a large size of effect. The d-value is based on the mean values and the pooled standard deviation of the estimates, i.e., it provides less robust results than our median-based tests.

The power of a test indicates the probability of detecting a significant relationship if there is one. This is highly relevant information for non-significant results. As pointed out in [14] and [15], many empirical studies have a too low power for relevant effects of size to draw meaningful conclusions on the basis of non-significant p-values. For non-significant tests we will, therefore, display the power for some relevant differences in mean values. The power tests are based on a significance level of 0.1. Notice that the power test only indicates the actual power since it is based on the, not completely met, assumption of normality of the distributions. It may, nevertheless, provide useful information to understand non-significant results.

In addition to the hypothesis tests described above, we explored the data for interesting relationships, i.e., we conducted a high number of tests and present only the most 'interesting' results. Statistical tests and p-values are misleading for data exploration. For this reason, we interpret the results based on data exploration only as starting points for further research.

In the statistical tests, we treat students and software professionals as samples from two different populations. However, since the populations are similar to each other we will use the results of the tests on one population in the discussion of the statistical significance of the results in the other population.

3. RESULTS

3.1 Estimates

The median number of described project activities was 10 for the students and 14 for the software professionals. The time used to understand the requirements specification and to estimate the development work was typically 30-40 minutes. There was no large difference between the students and the software professionals regarding time spent on the estimation task, or in how they described their estimation strategy. The participants decided themselves when the estimation task was completed.

Figures 1 and 2 show the estimated effort of the three groups for respectively the students and the software professionals. The median estimated effort of the students in the LOW group was 77 work-hours, and in the HIGH group 404 work-hours, i.e., more than five times as high! The median of the CONTROL group was

224 work-hours. The CONTROL group's median effort estimate was close to the median of 250 work-hours of actual effort used by other students on a very similar programming task. This indicates that the CONTROL group students had the most realistic[1] effort estimates.

The median estimated effort of the software professionals in the LOW group was 77 work-hours, and in the HIGH group 632 work-hours, i.e., more than eight times as high! The CONTROL group median effort estimate was 176 work-hours.

<Figure 1>

<Figure 2>

Table 1 shows the statistical values of the tests of the hypotheses H1-H3 for both populations.

<Table 1>

The absolute differences in median values are large for all groups and populations. Similarly, the size of effects, i.e., the differences adjusted for the variance of the estimates, are large or medium for all differences. The p-values are very low for all tests, except H2 and H3 when based on the population of software professionals. The p-values of these tests should, however, not be interpreted in isolation, since the statistical power of these tests is rather low and the population is similar to the student population. To get a power of about 0.8, i.e., a 80% probability that we would find a significant difference if one existed, we

---

[1] The notion of 'realistic' effort estimate is problematic. For example, in [6] it is suggested that different effort estimates lead to different software projects. Nevertheless, it is obvious that some estimates are more realistic than other. It is, for example,

would need about N=12 in H2 and N=30 in H3. The actual N was only 8. The large size of effect, the similarity to the student population and the low power imply that it is reasonable to assume that we would have found significant differences given a higher number of participating software professionals.

The number of tests is low, most of the p-values are very low and all statistical tests point in the same directions, i.e., the interpretation of the statistical results are not distorted by the number of tests. Overall, therefore, our findings support the hypotheses H1, H2 and H3.

The large variance of effort estimates within each group, depicted in Figure 1 and Figure 2, was no large surprise. It is realistic that different developers would use different amount of effort on the same task and that the specification would have been implemented differently.

There were only minor differences between the central values and spread of the effort estimates made by the students and the software professionals. This does not mean that we can join these two populations of subjects. The sample would then be dominated by very inexperienced software developers, i.e., not be a representative sample of the population of software developers. In addition, the assumed quality level for a finished system is probably different for students and software professionals, i.e., there would be a systematic difference in the interpretation of the estimation task.

3.2 Awareness

The participants in the LOW and HIGH groups were explicitly told not to consider customer expectation as valid input to the estimation process. Nevertheless, about 50% of the students and 40% of the software professionals in the LOW and HIGH groups indicated that their estimates of the most likely use of effort had been impacted by the customer expectations. None of the participants indicated a large impact. Typically, an impact of less than 20% of the estimate was indicated. Considering the large difference we measured, we interpret these findings as a strong support for hypothesis H4.

---

unlikely that 50% of the participants in the LOW group would complete the development using less than 77 work-hours, when the median effort of other similar students was about 250 work-hours.

A test of H5, i.e., that the participants who believed that they had been impacted were more impacted than the others, involves a division of the participants into the following sub-groups:

- HighNo. Participants in the HIGH group that answered that the customer expectation had no impact.
- HighYes. Participants in the HIGH group that answered that the customer expectation had an impact.
- LowNo. Participants in the LOW group that answered that the customer expectation had no impact.
- LowYes. Participants in the LOW group that answered that the customer expectation had an impact.

The number of participants in each group only enabled meaningful statistical analyses for the student populations.

Table 2 displays the results of the tests of different median estimates between the participants in the HighNo group and those in the HighYes group, and between the LowNo group and the LowYes group.

<Table 2>

Although the p-values point at no significant relationships, the low power of the tests means that we should not exclude small or medium differences in median values. For example, the probability to find a p-value of 0.1, given that the real difference in mean values is 50 work-hours or less, is only 40%. On the other hand, the power tests show that it is very likely that we would have found significant results if the size of effect had been as large as indicated by the test of H1 – H3. For example, assuming that the observed difference between the students' mean estimates of the CONTROL and LOW group (147 work-hours) is the actual difference, the probability of finding a p-value of 0.1 or less is about 95%. Hypothesis H5 is, therefore, supported for large differences, but we need more data to test for small and medium differences.

4. DISCUSSION

An impact of the customer's expectations may occur as a result of that, for example:

- The customer's high or low expectations bias the developers to think of the task as relatively easy or difficult. The development experience considered as relevant input to the estimation process is selected with this in mind.
- The developers use, to some extent, a top-down estimation process, even when the estimation instructions describe a bottom-up (activity based) estimation. The developers start the estimation process with an idea of what the total effort estimate should be. The starting point for this total effort estimate is the customer's expectation and the activity estimates are adjusted to fit the total effort estimate.

An interesting effect that is consistent with these two explanations and our findings, is the 'anchoring effect' described in several human judgement studies. In [16] this effect is described as *"the tendency of judges' estimates (or forecasts) to be influenced when they start with a 'convenient' estimate in making* *their forecasts. This initial estimate (or anchor) can be based on tradition, previous history or available data."* When the anchors are relevant, e.g., the customer's expectations are realistic, they lead to efficient and accurate estimates. On the other hand, when the anchors are irrelevant, as in our experiment, they may easily lead to biased estimates. The anchoring effect has been observed in many domains and tasks. [17, 18] provides an overview of studies on the anchoring effect and concludes that it may be difficult to protect against the effects of anchoring, because:

- incentives for accuracy seldom work,
- the anchor values often go unnoticed,
- irrelevant anchors have an impact even in situations where relevant anchors are present,
- judgements based on group processes may increase instead of reduce the anchoring effect and
- increased expertise does not remove the anchoring effect.

The results described in this paper seem to be similar to the results of other studies on anchoring effects, but may extend them regarding type of task (estimation of software development effort) and awareness analysis. We have been unable to find any study in the human judgement literature where the anchor is on the total value, e.g., total effort of a project, and the estimates are on sub-values, e.g., the effort of sub-activities. Other studies seem to focus on anchor values more directly related to the value to be estimated, e.g., a random number between 0 and 100 as an anchor for the estimate of a probability. In addition, we have found no other anchoring study in which the participants are asked to judge how much they were impacted.

Our analysis of the differences between students and software professionals suggests that computer science students with some programming experience are useful for studies on estimation of project effort. It is, however, important to be aware of the limitations. For example, the software professionals in our experiment had a larger experience base of activities necessary to complete a software project, and they knew more about the software quality expected by customers. A consequence of these differences was that although the impact of the customer expectation was large both for the students and the software professionals, the size and reasons for the impacts were to some extent different. A similarity in the resulting estimates of different populations is, therefore, not an evidence of equally similar underlying estimation strategies.

5. IMPLICATIONS

Two important implications for software practitioners, e.g., software project managers, are that irrelevant anchor values should be avoided or neutralised:

- *Avoid*. Ensure that the people estimating effort are not informed about the customer expectation regarding use of effort or cost. Avoid, if possible, that early estimates based on very limited information are presented to the customer.
- *Neutralise*. If avoidance is not possible, ensure that people have other anchor values in addition to the customer expectation. Historical data from earlier projects, estimation models and estimation checklists may become important anchor values neutralising, to some extent, the customer expectation.

There are limitations of these strategies, as well. Avoidance of the anchoring effect requires that we are aware of the impact of the customer expectation. Frequently, this may not be the case. In addition, results from other studies indicate that additional, relevant, information does not totally neutralise the anchoring effect, it only reduces it [17, 19, 20]. In our context this means that more relevant information, e.g., historical data, reduces the unwanted impact of the customer's expectations on the estimate of most likely use of effort, but does not remove it.

When possible, we should support the software developers with objective, historical data from many sources. This may reduce the bias toward too low effort estimates in situations where the customer expects an unrealistically low use of effort. The first challenge is to make the developers aware that they are impacted by irrelevant information and that there is a need to take the necessary precautions. We have used the results of this experiment to improve the estimation process of the organisation of the software professionals participating in this study. For example, the organisation now tries to hide the customer expectations from the developers in charge of the estimates and to postpone the effort estimation until enough relevant information is available. We believe that the organisation soon will see the results of these actions.

6. THREATS TO VALIDITY

The experimental situation is in many ways artificial, and there are consequently several threats to the validity of the results.

The participants knew that they would not have to develop the system. Therefore, their motivation for being realistic may be lower than for software developers providing effort estimates in real projects. However, investigating the activities specified to derive the estimates, the time used on the estimation task and the internal consistency of the activity based effort estimates, we found that the estimation task was taken seriously and perceived to be realistic.

The participants may have assumed that there were no real customer expectations in the experiment. The most likely consequence of this is, however, that we should expect even stronger impacts in more realistic situations with real customer expectations.

The participants may have interpreted the customer expectation as information about the quality of the software as expected by the customer. In some cases this is a rational estimation behaviour. However, the customer expectation regarding project cost is frequently not a reliable indicator of quality expectation. Instead, we believe, expectations of low cost are frequently related to lack of knowledge about the implication of the requirement specification combined with a desire to pay as little as possible, i.e., a part of the project negotiation or bidding process.

7. CONCLUSIONS

The experiment described in this paper indicates that customer expectations can have a surprisingly large impact on software development effort estimates, even when the estimators are told to disregard this information. The awareness of this impact seems low. Similar impacts, known as anchoring effects, have been reported in a number of studies.

The results of our study may explain a significant part of the effort underestimation in industrial software projects. In particular, the customer expectations may have an unwanted impact on the effort estimates when the software development organisation has communicated an early, very optimistic effort estimate to the customer. Simply knowing about this early estimate can be a major obstacle for a subsequent realistic effort estimate, even when the subsequent effort estimates are based on much more knowledge than was the early estimate, and the involved people are aware the very low realism of this early estimate.

ACKNOWLEDGEMENTS

This research is funded by The Research Council of Norway through the industry-project PROFIT (PROcess improvement For the IT industry). We are grateful to Prof. K. H. Teigen, Department of Psychology, University of Oslo, Norway for useful comments and discussions.

References

1. Hihn, J. and H. Habib-Agahi. *Cost estimation of software intensive projects: A survey of current practices*. Proceedings of *International Conference on Software Engineering*. 1991: IEEE Comput. Soc. Press, Los Alamitos, CA, USA. p. 276-287
2. Heemstra, F.J. and R.J. Kusters, Function point analysis: Evaluation of a software cost estimation model. *European Journal of Information Systems*, 1991. **1**(4): p. 223-237.

3. Paynter, J. *Project estimation using screenflow engineering*. Proceedings of *International Conference on Software Engineering: Education and Practice*. 1996. Dunedin, New Zealand: IEEE Comput. Soc. Press, Los Alamitos, CA, USA. p. 150-159
4. Hill, J., L.C. Thomas, and D.E. Allen, Experts' estimates of task durations in software development projects. *International Journal of Project Management*, 2000. **18**(1): p. 13-21.
5. Asher, The ambiguos nature of forecasts in project evaluation: Diagnosing the over-optimism of rate-of-return analysis. *International Journal of Forecasting*, 1993. **9**: p. 109-115.
6. Jørgensen, M. and D.I.K. Sjøberg, Impact of software effort estimation on software work. *Journal of Information and Software Technology*, 2001. **43**: p. 939-948.
7. Lederer, A.L. and J. Prasad, A causal model for software cost estimating error. *IEEE Transactions on Software Engineering*, 1998. **24**(2): p. 137-148.
8. Abdel-Hamid, T.K. and S.E. Madnick, *Software project dynamics. An integrated approach*. 1991, New Jersey: Prentice Hall.
9. Abdel-Hamid, T.K. and S.E. Madnick, Impact of schedule estimation on software project behavior. *IEEE Software*, 1986. **4**: p. 69-75.
10. Jørgensen, M. and D. Sjøberg. *The importance of not learning from experience*. Proceedings of *European Software Process Improvement 2000 (EuroSPI'2000)*. 2000. Copenhagen. p. 2.2-2.8
11. Wonnacott, T.H. and R.J. Wonnacott, *Introductory statistics*. 5th ed. 1990: John Wiley & Sons.
12. Rosenberg, W.F., Dealing with multiplicities in pharmacoepidemiologic studies. *Pharmacoepidemiology and drug safety*, 1996. **5**: p. 95-100.
13. Cohen, J., *Statistical power analysis for the behavioral sciences (Rev. ed)*. 1977, New York: Academic Press.
14. Cohen, J., The statistical power of abnormal-social psychological research: A review. *Journal of Abnormal and Social Psychology*, 1962. **65**: p. 145-153.

15. Sedlmeier, P. and G. Gigerenzer, Do studies of statistical power have an effect on the power of studies? *Psychological Bulletin*, 1989. **105**: p. 309-316.

16. Armstrong, J.S., *The forecasting dictionary*, in *Principles of forecasting: A handbook for researchers and practitioners*, J.S. Armstrong, Editor. 2001, Kluwer Academic Publishers: Boston. p. 761-824.

17. Whyte, G. and J.K. Sebenius, The effect of multiple anchors on anchoring in individual and group judgment. *Organizational behaviour and human decision making*, 1997. **69**(1): p. 75-85.

18. Plous, S., *The psychology of judgment and decision making*. 1993: McGraw-Hill.

19. Bazerman, M.H. and M.A. Neale, *Negotiating rationally*. 1992, New York: Free Press.

20. Switzer, F.S. and J.A. Sniezek, Judgment processes in motivation: anchoring and adjustment effects on judgment and behavior. *Organizational Behavior and Human Decision Processes*, 1991. **61**: p. 250-261.

FIGURES AND TABLES

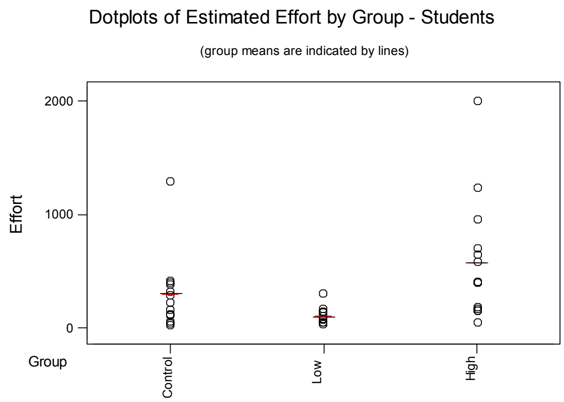

Figure 1. Effort estimates – students

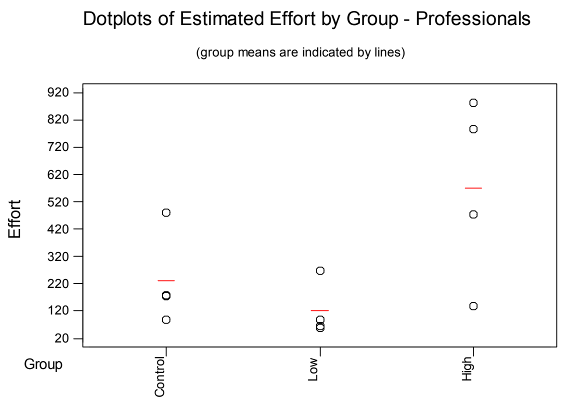

Figure 2. Effort estimates – software professionals

Table 1. Tests of hypotheses H1-H3

Hypothesis	**Population**	**N**	**Diff. median**	**p-value**	**Size of effect (d)**
H1 (HIGH vs LOW)	Student	25	327 work-hours	0.001	1.2 (large)
H2 (HIGH vs CONTROL)	Student	27	180 work-hours	0.07	0.6 (medium)
H3 (CONTROL vs LOW)	Student	24	147 work-hours	0.07	0.8 (large)
H1 (HIGH vs LOW)	Prof.	8	555 work-hours	0.04	1.8 (large)
H2 (HIGH vs CONTROL)	Prof.	8	456 work-hours	0.2	1.3 (large)
H3 (CONTROL vs LOW)	Prof.	8	99 work-hours	0.2	0.8 (large)

Table 2 Tests of hypothesis H5 (Students only)

Statistical test	**N**	**Diff. Median**	**p-value**	**Size of effect (d)**
Median of HighYes vs HighNo	13	95 (= 495 – 400)	0.7	0.0[2] (low)
Median of LowNo vs LowYes	11	55 (= 106 –51)	0.4	0.6 (medium)

[2] The size of effect is based on the mean values, which was exactly the same for these two groups.