An evaluation of the paired comparisons method for software sizing

Eduardo Miranda
Ericsson Research Canada
8400 Decarie Blvd.
Montreal, Quebec, Canada H4P 2N2
+1-514-345-7900
eduardo.miranda@lmc.ericsson.se

ABSTRACT
This paper evaluates the accuracy, precision and robustness of the paired comparisons method for software sizing and concludes that the results produced by it are superior to the so called "expert" approaches.

Keywords: software sizing, estimation methods,

Paired comparisons method, measurement methods, subjective estimates.

1 INTRODUCTION

Despite the existence of well known software sizing methods like Function Points [1] or the more recent Full Function Points [2] and Object Points [3], just to name a few, many practitioners and project managers continue to produce estimates based on ad-hoc or so called "expert" approaches.

Among the most common explanations given for not adopting more formal practices, we found:

- The lack of necessary information at the beginning of the project
- The specificity of the domain addressed by the method
- The effort and time required to apply them
- The need to introduce a foreign vocabulary to stakeholders without a software background

However, as figure 1.a shows, ad-hoc size estimates have problems of their own. Their accuracy and precision leave much to be desired. The problem is not an academic one, as uncertain size estimates automatically translate into questionable project budgets and schedules.

This paper presents the results of applying the paired comparisons method, a method used in the social sciences to measure soft issues, to software sizing.

The idea behind the method, is to estimate the size of *"n"* entities[1] by asking one or more experts to judge their relative largeness instead of to provide an absolute size value. By requiring multiple and explicit decisions about the relative size of every two entities, rather than a single comparison to some vague notion of size buried in the mind of the estimator, the paired comparisons method is able to improve both, the accuracy and the precision of the estimates as shown by Figure 1.b.

Although not new, the idea has received very little attention in the literature. Earlier work includes Target Software's Software Sizing Method [4], and more recently an article by Focal Point AB [5] where an instance of the method, called the Analytic Hierarchical Process, is used to prioritize requirements relative to their cost.

2 OVERALL APPROACH

As Figure 2 shows, first the artifacts to be sized are arranged according to their perceived largeness. Once this is done, the relative size of each of them with respect to all the others is assessed and recorded in a so called "judgement matrix". From the judgements made, a ratio scale is derived using a mathematical procedure. The absolute size of the entities is then calculated using the ratio scale and a reference value. Should the need arises, judgements could be reviewed for internal consistency. The following paragraphs briefly explain the steps and calculations required.

[1] These "entities" could be requirements, use cases, modules, features or objects or any other thing relevant to all stakeholders and for which it is possible to know the number of LOC, hours or any other magnitude that could later be used for planning purposes.

Artifacts to be sized

This is the list of of things: use cases, features, requirements, modules, etc., whose size the organization want to estimate.

It is very important that the size of the entities being estimated do not differ for more than one order of magnitude, as our ability ability to accurately discriminate size diminishes as the artifacts are further away[6,7].

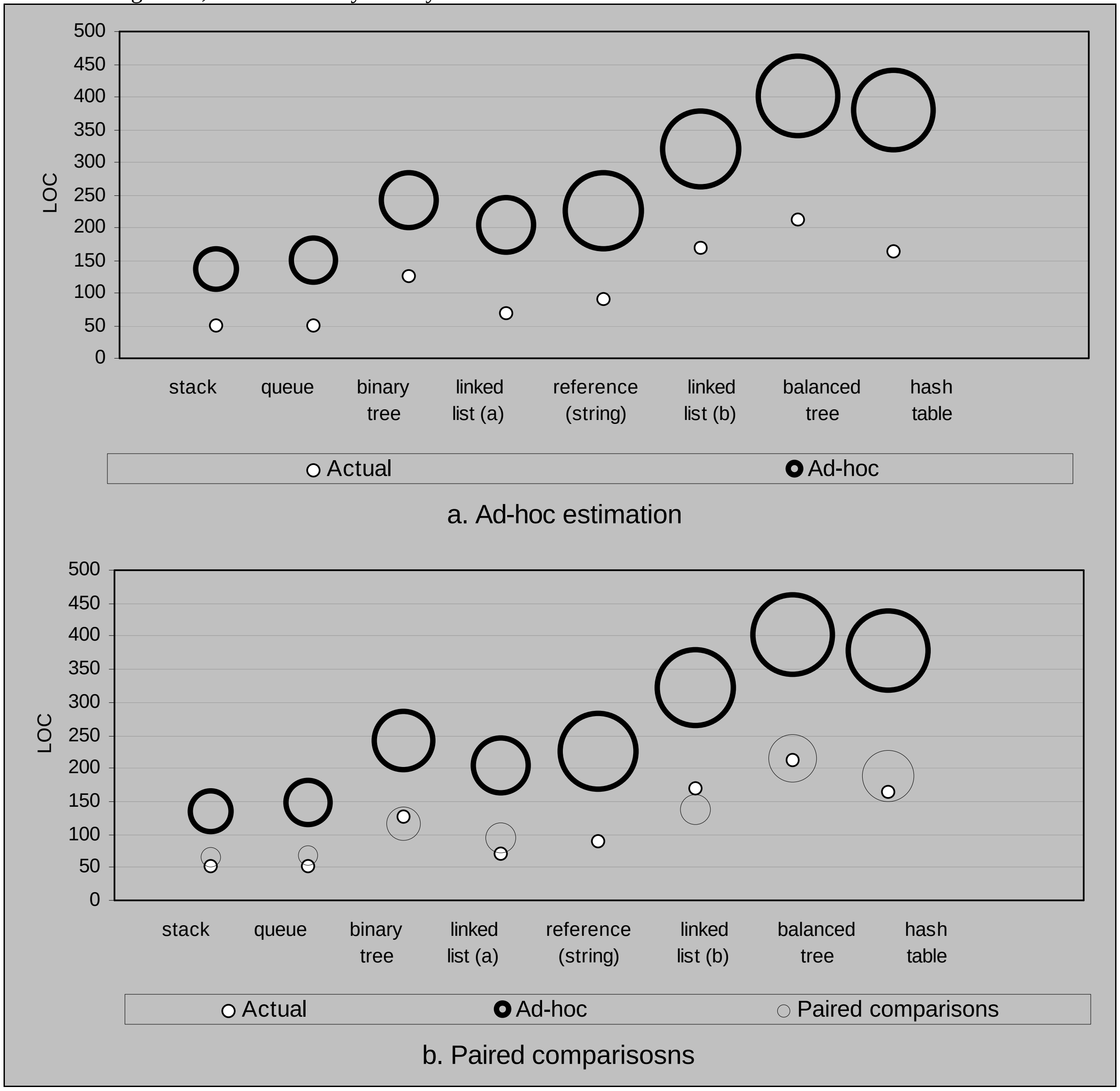
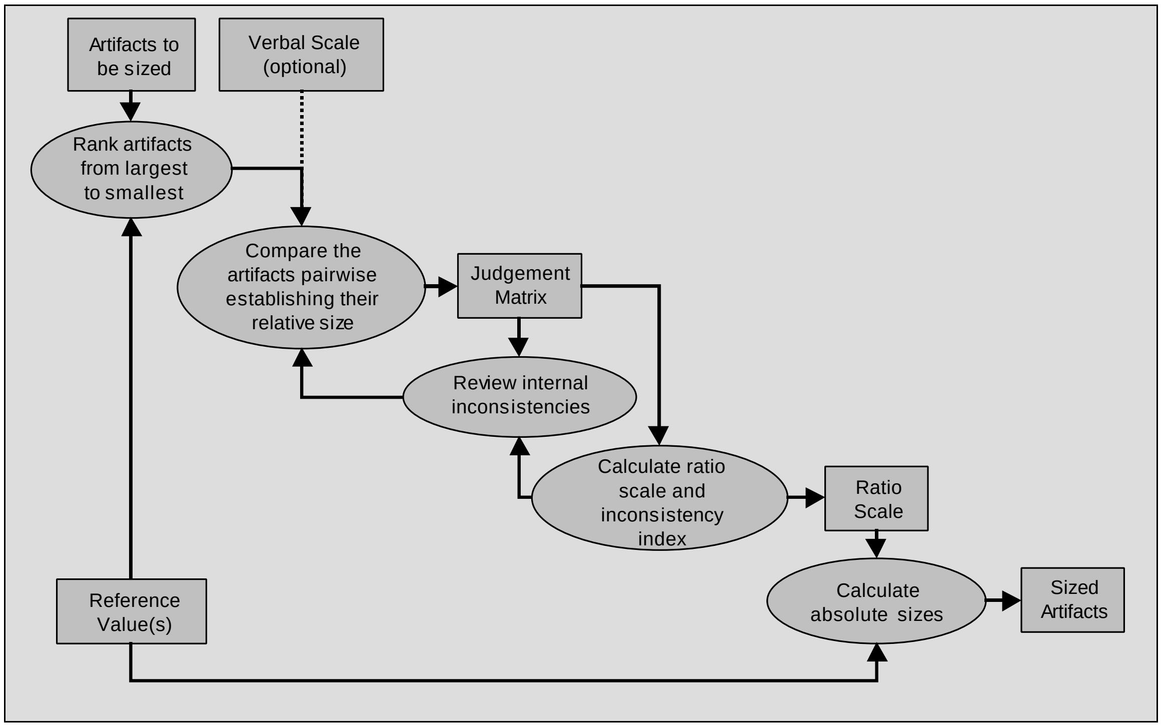


Figure 1 - Accuracy and precision of different estimation approaches



Artifacts to be sized
Verbal Scale (optional)
Rank artifacts from largest to smallest
Compare the artifacts pairwise establishing their relative size
Judgement Matrix
Review internal inconsistencies
Calculate ratio scale and inconsistency index
Ratio Scale
Calculate absolute sizes
Sized Artifacts
Reference Value(s)

Figure 2 – Estimation process

Judgement Matrices
A judgement matrix, is a square matrix, which captures the relative size of the artifacts being compared. The elements of the matrix are defined by Figure 3.

In practice, as shown by Table 1, all the judges need to do is provide estimates of the relative size of the $\frac{n*(n-1)}{2}$ upper elements of matrix. As all the other values could be derived from them.

$$A^{nxn} = [a_{ij}] = \begin{cases} a_{ij} = \dfrac{s_i}{s_j} & \text{, How much bigger (smaller) Entity}_i \text{ is with respect to Entity}_j \\ a_{ii} = 1 & \text{, Every entity has the same size as itself} \\ a_{ji} = \dfrac{1}{a_{ij}} & \text{, If Entity}_i \text{ is } a_{ij} \text{ times bigger (smaller) than Entity}_j\text{, then Entity}_j \text{ is } 1/a_{ij} \text{ times smaller (bigger) than Entity}_i \end{cases}$$

Figure 3 - Judgement matrix definition

Artifacts	**D**	**B**	**C**	**A**
D		4	6	7.5
B			1.5	2
C				2
A				

Element a_{ij} shows the relative size of entity *i* with respect to entity *j*. For example, $a_{12} = 4$ express the fact, that entity **D** has been judged four times bigger than **B**.

Table 1 – Judgement matrix example

Ranking according to size
Although not a mandatory step, arranging the artifacts in descending order according to their sizes, makes the rest of the process much easier.

As illustrated by Table 1, when the rows of a judgement matrix are sorted in descending order, the comparisons flow in one direction only, for example artifact “D” will be either equal or larger than any of the other entities against which is compared, it will never be smaller. Notice also, that within a row, the values to the left of any given column are always smaller or equal than those to the right.

While these properties are irrelevant from the mathematical point of view, they diminished the strain put on the judges by the large number of decisions required from them by the method.

The Reference Value(s)

The paired comparison method, requires the existence of at least one reference value whose size is known, for example from a previous development.

The reference value is first ranked and compared to every other artifact as would any of the entities to be sized. Later it is used to calculate the absolute size of elements.

The choice of a reference value is an important decision, as values in either extreme of the scale tend to amplify any bias that might affect the judgements. To minimize this risk, is better to choose as reference, an artifact that will divide the population being estimated in halves, or to use two references instead of one.

The Verbal Scale, How small is smaller, how big is bigger?

Although not an essential part of the methodology, having a shared understanding of how small is smaller and how big is bigger, helps reach consensus among participants in the sizing process.

A predefined value scale, keeps us from wasting time discussing values down to the second decimal, when our judgement error is one or two orders of magnitude bigger than that.

Quoting an earlier work from Ernest H. Weber, Saaty proposes to use a scale from 1 to 9 and their reciprocals, to pass judgment on the entities been evaluated. The equivalence between verbal expresions and relative sizes is depicted by Table 2.

A survey ran among some colleagues[2], suggests that the correspondence between size and verbal description in the software domain, is closer to the one provided by Table 3, rather than Saaty's.

The use of the verbal scale simplifies and speeds-up the estimation process without jeopardizing the accuracy of the results.

[2] Thirty people from different industries, countries and academia provided their input for this scale. This table might need to be reformulated if the entities being compared are further apart from each other.

Calculating a ratio scale and an inconsistency index

A ratio scale is a vector [r_1, r_2,…, r_n] in which each number r_i, is proportional to the size of entity *i*. An Inconsistency Index is a number which measures how far away are our judgements from being perfectly consistent[3].

There are many ways to calculate ratio scales and inconsistency indexes. In this paper we will use the Crawford & Williams' Geometric Mean Procedure [4] because of its simplicity and good results.

Calculating absolute sizes

Giving a ratio scale [r_1, r_2,…, r_n], the absolute sizes of the artifacts being estimated are calculated using the expression in Figure 5.

3 ACCURACY AND PRECISION

In the theory of measurements, accuracy is defined as how close the measurements are to the true value or known input, and precision as the ability to reproduce a set of measurements with a given accuracy. The closer the individual measurements are to one another, the higher the precision.

As with any other measurement instrument, it is important to understand the accuracy and precision that could be expected when using the paired comparison method.

In order to evaluate the method's performance I ran four experiments with the results illustrated by Figures 6, 7, 8 and 9. on them, the area of the circles corresponds to the standard deviation of the observations.

The purpose of the first experiment was to evaluate the robustness of the method under the conditions described bellow using controlled inputs.

1. Reference[4] values, E_i, in KSLOC: [50, 30 ,25, 24, 24,10, 9, 8, 5, 5];
2. Estimation with simulated judgements (E_i / E_j)+ ε, where ε is a random variable with normal distribution, $\mu = 0$ and $\sigma = .15$ (E_i / E_j). In other words, assuming that 68% of the time, the judgements are within 15% of their true value;
3. Idem 2, but assuming the judgements are within 30% of their true value;

[3] A perfectly consistent judgement matrix is one in which all its elements satisfy the condition $a_{ij}a_{jk} = a_{ik}$ for all *i, j, k.*

[4] In a real situation, these are the values we want to estimate.

4. Estimation resulting from mapping the judgements in 3 above onto the verbal scale for the software domain; and

5. Estimation resulting from mapping the judgements in 3 above onto Saaty's verbal scale.

The results of this experiment show that the method produces consistent results even on the presence of large judgement errors, as long as they are not biased.

Definition	**Explanation**	**Relative Value**	**Reciprocal**
Equal size	The two entities are roughly the same size.	1	1
Slightly bigger (smaller)	Experience and/or judgement recognize one entity as being somehow bigger (smaller)	3	.33
Bigger (smaller)	Experience and/or judgement recognize one entity as being definitely bigger (smaller)	5	.2
Much Bigger (smaller)	The dominance of one entity over the other is self-evident. Very strong difference in size (smaller)	7	.14
Extremely bigger (smaller)	The difference between the entities being compared is of an order of magnitude	9	.11
Intermediate values between adjacent scales	When compromise is needed	2, 4, 6, 8	.5, .25, .16, .12

Table 2 - Saaty's verbal scale

Definition	**Explanation**	**Relative Value**	**Reciprocal**
Equal size	$E_i / E_j \leq 1.25$ (0~25%)	1	1
Slightly bigger (smaller)	$1.25 < E_i / E_j \leq 1.75$ (25 ~ 75%)	1.25	.80
Bigger (smaller)	$1.75 < E_i / E_j \leq 2.275$ (75 ~ 275%)	1.75	.57
Much Bigger (smaller)	$2.275 < E_i / E_j \leq 5.75$ (275 ~ 575%)	4	.25
Extremely bigger (smaller)	$5.75 < E_i / E_j \leq 10$ (575 ~ 1000%)	7.50	.13

Table 3 - Verbal scale for the software domain

$$\text{(a)} \quad n_i = \sqrt[n]{\prod_{j=1}^{n} a_{ij}} \qquad \text{(b)} \quad r_i = \frac{n_i}{\sum_{l=1}^{n} n_l}$$

$$\text{(c)} \quad InconsistencyIndex = \frac{\sqrt{\sum_{i=1}^{n}\sum_{j>i}^{n}\left(\ln a_{ij} - \ln\frac{n_i}{n_j}\right)^2}}{\frac{(n-1)(n-2)}{2}}$$

Figure 4 - Crawford & Williams procedure, (a) geometric mean, (b) ratio scale, (c) inconsistency Index

$$Size_i = \frac{r_i}{r_{reference}} * Size_{reference}$$

Figure 5 – Entity *i* absolute size

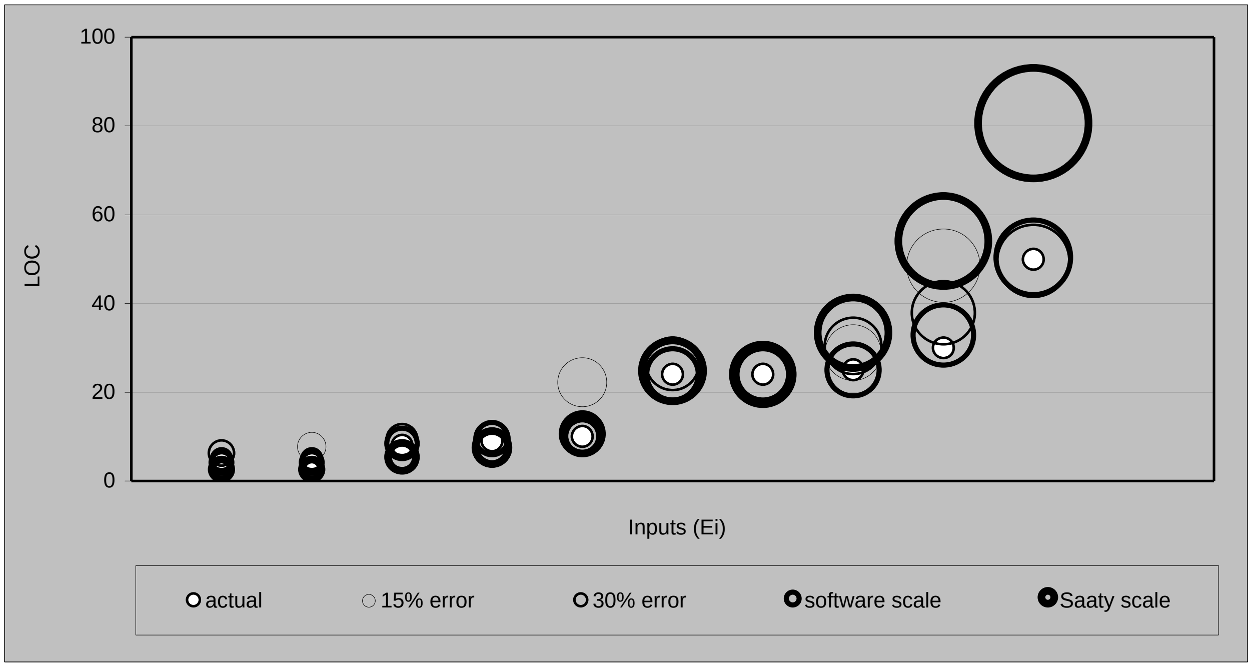


Figure 6 – Results using simulated inputs

The second experiment was based on a survey of 30 professionals and graduate students, who were asked to assess the absolute or relative size of the data structures[5] mentioned in the chart, using one of three methods:

1. By providing the absolute size of the data structures based on the judge's best knowledge, this is referred as the Ad-hoc method;
2. Using paired comparisons and rating the relative largeness of one data structure with respect to the others using a number; and
3. Using paired comparisons but this time rating the relative largeness of one data structure with respect to the others using the verbal scale for the software domain.

[5] Judges were provided with a brief specification, and assumptions about the language to use, amount of error checking and comments, in order to have a common understanding for the estimation. The actuals are an average of actual programs extracted from libraries written in C, Ada and C++ and adjusted for the functionality defined in the questionnaire.

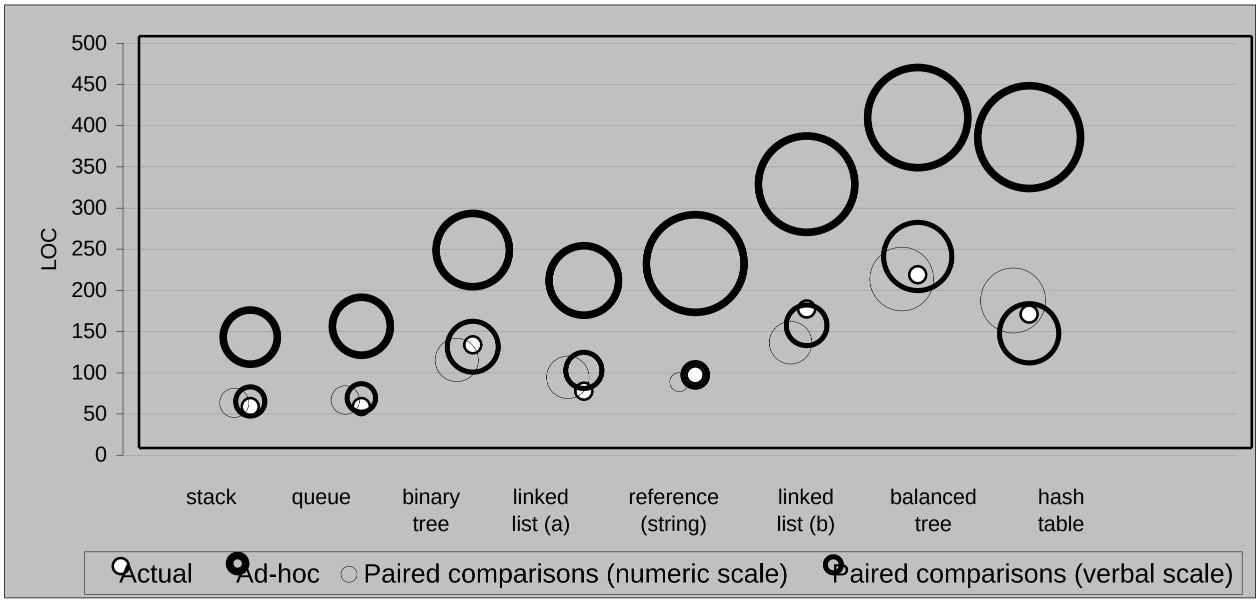


Figure 7 – Estimation results

From the results we can conclude, that there are significant[6] differences between the results obtained using paired comparisons versus the ad-hoc approach. They also show, that there are not significant differences between the use of the numeric and the verbal scales.

The third experiment consisted on the estimation of the area enclosed by different geometrical figures, the purpose this time was to study the performance of the method when applied to other kind of subjective estimations. In this case, as the ratio of the largest to the smallest shape exceeded an order of magnitude, the Saaty's scale was preferred to the verbal scale for the software domain, as this last one tended to understimate the largest figures.

By looking at Figure 8, it is possible to observe the individual estimates for each geometrical figure. The precision of the different estimation methods is reflected by the closeness of the points, the closer they are the more precise the method. The accuracy is shown by the position of the estimates relative to their respective reference point, shown with a cross in the chart. It is interesting to notice that in all the methods, accuracy and precision tend to deteriorate as the points being estimated are further away in terms of their relative magnitude. This seems to confirm Saaty's assertion, that comparing items whose size differ by more than 10 is meaningless.

4 CONCLUSIONS

The paired comparisons method for software sizing is specially well suited for the early stages of a development project or during feasibility studies, when the knowledge available to the members of the project team, is mostly qualitative.

The results observed so far show that the method is robust and perform consistently. Further experimentation is necessary to establish the validity of the verbal scale for the software domain and to verify that the potential application to the method to other software engineering problems requiring subjective estimates.

[6] In order to avoid making assumptions about the underlying distribution of the answers to the questionnaire, a non-parametric *t*-Test for samples with unequal variance was used with the results showing significance at the 95% level of confidence.

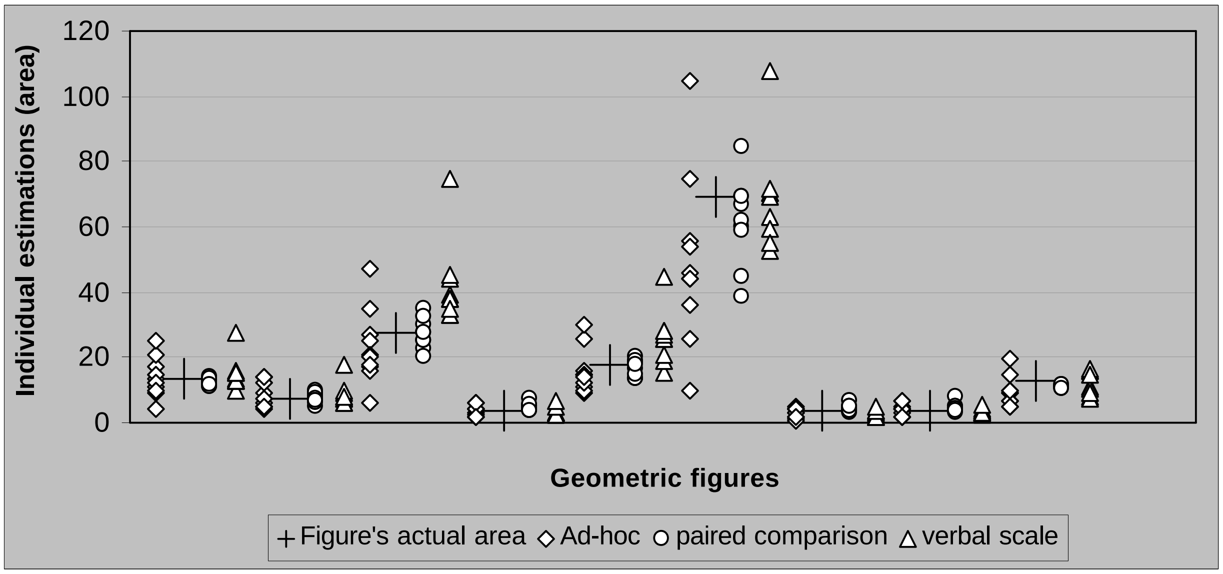


Figure 8 – Estimating the area of geometrical figures

References

1. Albrecht, A & Gaffney, J., 1983, Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, IEEE Transactions on Software Engineering, Vol. SE-9, No. 6,pp 639-648
2. COSMIC-Full Function Points - Release 2.0, September 1999
3. Object Points, Arlene F. Minkiewicz, PRICE Systems, Lockheed Martin Corporation, http://www.pricesystems.com/foresight/arlepops.htm
4. G. Bozoki, An expert judgement based software sizing model, Lockheed Missiles & Space Company and Target Software
5. J. Karlsson & K. Ryan, A Cost-Value Approach for Prioritizing Requirements, IEEE Software, September/October 1997
6. E. Miranda, Sizing Software Using The Paired Comparisons Method, Proceedings of the 9th International Software Measurement Workshop, 1999
7. T. Saaty, Multicreteria Decision Making: The Analytic Hierarchy Process, RWS Publications, 1996
8. G. Crawford and C. Williams, The Analysis of subjective judgement matrices, Rand Corporation, 1985