Sizing User Stories Using Paired Comparisons

© Eduardo Miranda, Pierre Bourque, Alain Abran
École de technologie supérieure

Abstract
Agile estimation approaches usually start by sizing the user stories to be developed by comparing them among themselves. Different techniques, with varying degrees of formality are used to conduct the comparisons – plain contrasts, triangulation, planning poker and voting. This article proposes the use of a modified paired comparison method in which a reduced number of comparisons is selected according to an incomplete cyclical design. Using two sets of data the authors show that the proposed method produces good estimates even when the number of comparisons is reduced by half those required by the original formulation of the method. The authors also consider the possibility of extending the use of the technique to maintenance requests and other fine grained development.

Keywords
Agile estimation, triangulation, estimating by analogy, paired comparisons, software sizing, effort estimation, project planning, incomplete cyclic designs

1. Introduction
Agile estimating approaches typically comprise three steps: 1) comparing user stories to be developed to each other with the purpose of establishing their relative size, 2) converting the size estimates to lead times using an assumed team productivity; and 3) re-estimating the project lead-times using the actual team's productivity once this becomes known after two or three iterations.

Comparisons among user stories take the following form: “This story is like that other story, so its size must be roughly the same” or “This story is a little bit bigger than that other story that was estimated at 4 so its size should be around 5”. Four and five in the previous sentence are called “story points”, which are numbers in an interval scale purportedly proportional to the effort it would take to develop each story based on its perceived size and complexity. A 6-point user story is expected to require about twice as much effort as a 3-point user story. The degree of structure in the comparison process ranges from the ad-hoc comparison of any two user stories, comparison of a user story with other two – triangulation, and a number of Delphi [1] like techniques such as the planning poker [2]. To avoid wasting time discussing insignificant differences between user stories, the use of a Fibonacci or power series is sometimes recommended such as if the difference between two user stories is not as large as a following term in the series, the two user stories are assumed of the same size [3]. Sometimes organizations choose to express the size of the user stores in “ideal days”, which is a measure of how long it would take the team to develop the story if that is all it worked on. As we found the concept of measuring size in days misleading we will not use it in the rest of the paper.

The project lead time is calculated using the concept of “velocity” which is a proxy for the productivity of the team. At first, the velocity is estimated or taken from a previous project, but as work progresses it is measured by tallying the number of story points completed during the counting period. Velocity is measured in story points per iteration or story
points per month. As an example, if the current team velocity is 30 story points per
month it will take the team 2 months to deliver 60 story points worth of user stories.

This paper addresses only the first step of the process: comparing user stories to each
other with the purpose of establishing its size.

2. Agile estimation and triangulation

Triangulation is defined in the agile literature as the process of establishing the size of a
user story relative to two other user stories with the purpose of increasing the reliability1
of the estimate [2]. When using triangulation the comparisons sound something like this
"I'm giving this story 2 points because it feels like it's somewhere between that 1-point
story and that 4-points story". Despite its intuitive appeal, triangulating is not as simple
as the sentence above makes it appear. First there is the problem of consistency which
can be mathematically expressed as:

\[ a_{ij} \times a_{jk} = a_{ik} \quad \forall i, j, k \in n \]

Equation 1 is read as follows: if user story \( i \) is \( a_{ij} \) times bigger than user story \( j \) and
user story \( j \) is \( a_{jk} \) times bigger than user story \( k \) then user story \( i \) must be \( a_{ij} \times a_{jk} \) times
bigger than user story \( k \). This is important because lack of consistency among
triangulations leads to inaccurate estimates.

Second, which two user stories should you choose as reference points? Does the choice
affect the result?

The triangulation process can be visualized, see figure 1, by arranging the user stories in
a circular pattern and by linking the stories being compared. Given \( n \) user stories to be
estimated there are \( n(n-1)(n-2)/2 \) possible configurations or designs that can be
evaluated but not are all equally good. A good design must satisfy two properties:
balance and connectedness [4-6]. A design is considered balanced when every user
story appears in as many comparisons as any other user story. This assures that one
user story is not overtly influencing the estimation while others are underrepresented.
Connectedness implies that any user story is compared, directly or indirectly, to every
other user story. An unconnected graph is undesirable because the size of some user
stories relative to others would be completely indeterminate. Figure 1.b illustrates the
problem: the user stories in the lower subset are never compared against those in the
upper subset so each subset could be accurately sized in itself but completely offset with
respect to the other.

The number of times a user story appears in a comparison is called the replication factor
(\( r \)) of the design. The replication factor in all the designs shown in Figure 1 is two.
Balance and connectedness are necessary but not sufficient conditions for a good
estimation. As shown by Burton [4], a low replication factor such as the one used in the
triangulation approach (\( r = 2 \)) is very sensitive to errors in judgments and thus tends to
produce unreliable results. In his experiments Burton found that the correlation (\( \rho \))

---

1 A reliable sizing method will yield estimates that are accurate, that is close to close to their true
value, and precise, that is they will not depend on the choice of user stories being compared.
2 The comparison can go both ways, i.e. replacing bigger for smaller.
between the actual and the estimated values using triangulation ranged from a low of 0.46 to a high of 0.92, with a mean value of 0.79. Similar variability was found by the authors using two sets of data and is explained later.

![Diagram of triangulation designs](image)

**Figure 1** Three possible triangulation designs out of the $n(n-1)(n-2)/2$ possible ones

Note 1: 'a' and 'c' are good designs, 'b' is not, as it consists of two disjoint subgraphs.

### 3. Using more than two user stories as reference points

The obvious solution to the low reliability of the simple triangulation design is to increase the number of points of reference. This not only begets the question of how many points are good enough but also makes the problem of dealing with inconsistent judgments more acute.
To solve these problems we propose to use the mathematics of the paired comparison method [7-9] to deal with the estimation and to use incomplete cyclic designs to identify which user stories to compare with which to reach a desired accuracy.

The rest of the paper is organized as follows. Section 4 explains the basic paired comparison method, section 5 the reduction of the number of comparisons by generating incomplete cyclic designs, section 6 an empirical verification to evaluate the accuracy and precision of the resulting estimates and section 7 a summary.

4. Paired comparisons basics

4.1. Overview

The idea behind the paired comparison method is to estimate the size of \( n \) user stories by asking one or more developers to judge their relative largeness rather than to provide absolute size values. Later, by using a user story with a known number of story points as reference, the sizes of all the other user stories are calculated. The process is shown in Figure 2. This process is called “full factorial” because it compares all user stories (factors) against each other.

![Figure 2 The full factorial paired comparison process](image)

Note 1: Every user story is compared to every other.

Note 2: The optional verbal scale allows comparing the user stories using an ordinal scale by labeling the comparison of two user stories with adjectives such as “equal”, “a little bigger”, “much bigger”, etc.

4.2. Comparing User Stories Pairwise

Developers start the process by judging the relative largeness \( (a_{ij}) \) of each user story against every other user story and recording these values in a matrix called the judgment matrix <2>. 

```
\[
A^{\text{row}} = \begin{cases}
1 & \text{if user story } i \text{ is } a_i \text{ times bigger (smaller) than user story } j
\\
\frac{1}{a_i} & \text{if user story } j \text{ is } a_i \text{ times smaller (bigger) than user story } i
\end{cases}
\]

\[a_i = \frac{\text{sp}_i}{\text{sp}_j}\] How much bigger (smaller) user story \(i\) is with respect to user story \(j\).

Every user story has the same size as itself.

\[a_i = \frac{1}{a_i}\] If user story \(i\) is \(a_i\) times bigger (smaller) than user story \(j\), then user story \(j\) is \(1/a_i\) times smaller (bigger) than user story \(i\).

\(sp_i\) and \(sp_j\) are the yet unknown numbers of story points for user story \(i\) and user story \(j\) to be derived from the \(a_{ij}\) judgments. Notice that only the comparisons corresponding to the upper-diagonal matrix have to be made since the \(a_{ji}\) are the reciprocals of the \(a_{ij}\).

4.3. Calculating the Size, the Inconsistency Index and the Standard Deviation

Once the \(a_{ij}\) judgments have been recorded, the mean relative size (\(mrs_i\)) of user story \(i\) is calculated as the geometric mean [10, 11] of the \(i\)th row <3> of the judgment matrix. The size in story points of each user story is then computed by multiplying its \(mrs_i\) by the normalized size of the reference user story <4>. For a more detailed description of the method refer to [7, 8].

\[mrs_i = \left( \prod_{j=1}^{n} a_{ij} \right)^{\frac{1}{n}}\] <3>

\[sp_i = \frac{\text{sp}_{\text{reference}}}{mrs_i} \times mrs_i\] <4>

As judgments will seldom be perfectly consistent, Crawford [11] and Aguaron [12] suggest the use of <5> as an unbiased estimator of the variance of the inconsistencies of the judgment matrix. The larger the inconsistencies between comparisons, the larger the variance will be. The square root of <5> is called the Inconsistency Index <6> of the judgment matrix.

\[
\sigma^2 = \frac{\sum_{i=1}^{n} \left( \ln a_{ij} - \ln \frac{\text{mrs}_i}{\text{mrs}_j} \right)^2}{n(n-1) - (n-1)}
\] <5>

\[
\text{InconsistencyIndex} = \sqrt{\sigma^2} = \sqrt{\frac{2\sum_{i=1}^{n} \left( \ln a_{ij} - \ln \frac{\text{mrs}_i}{\text{mrs}_j} \right)^2}{(n-1)(n-2)}}
\] <6>
While the Inconsistency Index gives an overall idea of the quality of the judgments, it is a quantity that is difficult to interpret. A much better alternative is to present the estimator with a range of values within which the estimate will lay for a given degree of inconsistency.

Assuming that each user story contributes equally to the total variance of the judgments, \( \sigma_i^2 \), allows us to write equation <7> where \( \sigma_i^2 \) is the individual inconsistency contributed by each user story and \( n - 1 \) the degrees of freedom of the statistic. We use \( n - 1 \) in the denominator instead of \( n \) because the reference user story does not introduce any inconsistency (its size is known – this is why we are using it as reference). The standard deviation of the size of each user story could then be calculated as the product of its estimated size by its individual inconsistency <8>.

\[
\sigma_i^2 = \sum_{i=1}^{n-1} \sigma_i^2 = (n-1)\sigma_i^2
\]

\[
\sigma_{sp} = \sqrt{sp} \times \sigma_i
\]

Other approaches to calculate the standard deviation of the size exist. Hihn [13] propose the use of a triangular distribution where the developer or estimator judges the relation between two user stories in terms of the best case, most likely and worst case scenarios, but given the rather large number of user stories included in a typical project we found this to be too taxing.

### 4.4. Reviewing Inconsistencies

At this point the estimator will use the inconsistency index or the estimates’ range as guidance to decide whether they are good enough or revise all or some of his or her judgments to reduce the inconsistencies.

By simulating a large number of judgments matrices and comparing it to perfectly consistent ones, Aguaron [12] determined that for sets with four or more data points, an inconsistency index of less or equal than 0.35 would produce satisfactory results in most cases. To give the reader an idea of its meaning, an inconsistency index of 0.35 would, under the assumption that all user stories contribute equally to it, result in a size range of \( \pm 3.2\% \) with 15 user stories being compared. If fewer user stories were compared the size range would be wider. If more stories were compared, the interval would be narrower.
4.5. A Numerical Example

Suppose we wanted to estimate the story points of three user stories called A, B and C using as reference a fourth labeled D whose number of story points had been judged to be 5. If we judge the size of A being three times that of B, twice that of D and half that of C and then we assess B to be roughly once and a half times D and a quarter of C and C as being five times bigger than D, the resulting matrix is:

\[
\begin{pmatrix}
A & B & C & D \\
1 & 3.0 & .50 & 2.0 \\
.33 & 1 & .25 & 1.5 \\
2.0 & 4.0 & 1 & 5.0 \\
.50 & .70 & .20 & 1 \\
\end{pmatrix}
\]

Only the relative largeness of the upper diagonal elements of the matrix need to be judged as all the other values can be derived using the definitions in <2>. Applying equation <3>, the mean relative size of each user story is:

\[
m_{RS} = \begin{pmatrix} 
2.34 \\
0.59 \\
1.41 \\
0.50 \\
\end{pmatrix}
\]

with an InconsistencyIndex of 0.94.

The story points of each user story are calculated using equation <4> using \(sp_A = 5\) as reference:

\[
sp_i = \begin{pmatrix} 
23.0 \\
5.9 \\
13.9 \\
5.0 \\
\end{pmatrix}
\]

The total size for the project is 47.8 story points or 42.8 if the reference user story is excluded. Notice that the reference user story is not the same as a “gold standard” [3] against which all other user stories are compared in exclusivity.

According to <8>, an InconsistencyIndex of 0.94 with 4 user stories being estimated will yield a size range of ± 51%, <9>.

\[
\frac{0.94^2}{\sqrt{4-1}} \times 100 = 51\%
\]

As the estimator is not happy with such a range he uses equation <1> to scan for the largest inconsistent judgments, see Figure 3, and decides to revise his estimates of the
relationships between B and C, between A and D and between B and D as stated in Error! Reference source not found.

\[
\begin{array}{c|cccc}
& A & B & C & D \\
A & 1 & 3.0 & .50 & 8.0 \\
B & 33 & 1 & 1.0 & 2.0 \\
C & 2.0 & 1.0 & 1 & 5.0 \\
D & 12 & .50 & .20 & 1 \\
\end{array}
\]

Figure 3 Tool interface for detecting the most inconsistent judgments.

Note 1: Each time the “Analyze” button is pressed a new triad is displayed.

Note 2: The “spinner” is used to adjust the inconsistency threshold used in the scan.

These changes reduce the Inconsistency Index from 0.94 to 0.45. The lower value indicates a more consistent evaluation of the relative size of the user stories. As the new Inconsistency Index of 0.45 will result in a size range for each user story of ±11% the estimator decides to accept the results.

The new total size for the project is 97.4 ±9.13 story points with individual sizes and standard deviations of:
5. Reducing the number of comparisons with Incomplete Cycle Designs (ICD)

5.1. Overview
The full factorial paired comparisons method calculates the user stories’ size by triangulating each of them with all other user stories in the project and measuring the consistency of the judgments made but it does not scale-up as the number of comparisons required grows with the square of the number of user stories being estimated.

To solve this problem several authors [5, 6, 14] have proposed the use of Incomplete Cyclic Designs (ICD) to select a subset of the \( n(n-1)/2 \) comparisons required by the full factorial method. Incomplete cyclic designs owe their name to the way user stories are selected for comparison. Starting with a first user story successive comparisons are selected in a cyclic fashion using arithmetic modulo \( n \). This is further developed in the following sections.

Table 1, shows the results of a series of experiments conducted by Burton [4] to evaluate the impact of a reduced number of comparisons in the reliability of the measurements. The experiment consisted in evaluating the correlation between the results of a full factorial estimation with the results of a number of fractional (ICD) designs with different replication factors. The closer to one the correlation between the full factorial and the fractional designs and the lower the spread between the lowest and the mean correlations, the more accurate and precise the values estimated using the ICD were.

The data set used in the experiment consisted of 21 different concepts for which the test subjects needed to quantify their semantic similarity. The full factorial design required 210 comparisons. The experiment showed that a fractional design with only 63 comparisons displayed a high correlation (0.80 to 0.96) between the similarities estimated by the two different methods.
### Table 1 Results from Burton experiments

<table>
<thead>
<tr>
<th>Number of comparisons in which each concept was included (r)</th>
<th>Number of comparisons with respect to the complete design</th>
<th>Number of comparisons in full factorial design</th>
<th>Number of comparisons in the ICD</th>
<th>Lowest correlation with results from complete design</th>
<th>Mean correlation with results from complete design</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10%</td>
<td></td>
<td>21</td>
<td>.46</td>
<td>.79</td>
<td>Mean correlation is acceptable but worst case is too low</td>
</tr>
<tr>
<td>4</td>
<td>20%</td>
<td></td>
<td>42</td>
<td>.58</td>
<td>.95</td>
<td>Mean correlation and worst case are acceptable</td>
</tr>
<tr>
<td>6</td>
<td>30%</td>
<td>210</td>
<td>63</td>
<td>.80</td>
<td>.96</td>
<td>Mean correlation and worst case are acceptable</td>
</tr>
<tr>
<td>8</td>
<td>40%</td>
<td></td>
<td>84</td>
<td>.97</td>
<td>.98</td>
<td>Almost as good as the complete design</td>
</tr>
</tbody>
</table>

#### 5.2. The fractional paired comparison process.

The term fractional is used to describe the fact that the method uses a fraction of all the comparisons required by the full factorial design.

The fractional paired comparisons method requires that: (1) we decide which comparisons to make and (2) that we compensate for the missing values. This adds to the original process two new activities: Generate Incomplete Cyclic Designs and Impute Missing Values, see Figure 4, which are explained in later sections.
5.3. Generating Incomplete Cyclic Designs (ICD)

The proposed Incomplete Design Cycle (ICD) construction process starts by arranging, in a random order, the user stories along a circle and joining adjacent user stories with a line. Each line corresponds to a comparison.

The design generated this way - see Figure 5.a - consists of a total of 7 comparisons with each user story appearing in two comparisons, one with the user story to its left and the other with the one to its right.

\[ s \] , the design’s distance, is the minimum number of hops along the circle needed to reach the stories being compared.

In Figure 5.a, \( r = 2 \) and \( s = 1 \).

Additional designs, see Figures 5.b and 5.c are generated by increasing the distance between compared user stories. ICD with a higher replication factors are obtained by merging simpler designs as shown by Figure 5.d which results from the juxtaposition of the 5.a and 5.c designs.
Figure 5 Four different incomplete cyclic designs.

Note 1: Design a, b & c are created by increasing the distance (s) between user stories.

Note 2: Design d is the result of merging designs a & c.
To operationalize the generation of ICDs we use an adjacency matrix $G^{n \times n}$ and the algorithm in figure 6. Table 2 shows the matrix representation of the designs in Figure 5.

1. Number the user stories to be sized $0, 1, 2, \ldots, n - 1$

   Create a matrix $G^{n \times n}$; initially all the elements of the matrix are False

   $G^{n \times n} = \begin{cases} g_{ij} = True & \text{if the comparison } a_j \text{ is to be included in the design} \\ g_{ij} = False & \text{otherwise} \end{cases}$

2. For $s = 1$ to $\lceil r / 2 \rceil$

   ‘$r$ is the desired replication factor

   2.1. For $i = 0$ to $n - 1$

   2.1.1. $j = (s + i) \mod n$

   2.1.2. If $Not(g_{ji})$ then $g_{ji} = True$ ‘this check prevents the inclusion of an

   2.2. Next $i$

   ‘element if its reciprocal is already included

3. Next $s$

---

**Table 2 Matrix representation of the designs in Figure 5**

<table>
<thead>
<tr>
<th>User story</th>
<th>A</th>
<th>C</th>
<th>F</th>
<th>D</th>
<th>G</th>
<th>B</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>a, d</td>
<td>b</td>
<td>c, d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>a, d</td>
<td>b</td>
<td>c, d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>a, d</td>
<td>b</td>
<td>c, d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>a, d</td>
<td>b</td>
<td>c, d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>c, d</td>
<td>a, d</td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>b</td>
<td>c, d</td>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>a, d</td>
<td>b</td>
<td>c, d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) $r = 2, s = 1$; (b) $r = 2, s = 2$; (c) $r = 2, s = 3$; (d) $r = 4$

---

5.4. **Imputing Missing Values**

Before we can calculate the size of the user stories, we need to impute the missing values of the judgment matrix, those corresponding to the comparisons that were skipped by the design, with a representative value. Notice that the assignments $a_{ii} = 1$ and $a_{ji} = 1 / a_{ij}$, the elements on the principal diagonal and the reciprocal of the judgments respectively, are not missing values and should be made before the imputing calculations.
As the comparisons to be skipped were selected at random by the ICD construction procedure, we can impute them with the mean value [15, 16] of the row to which they would have belonged using the algorithm in figure 7.

Assume the existence of a judgment matrix $A^{xxx}$ filled according to the ICD represented by the matrix $G^{xxx}$ whose missing $a_{ij}$ values are zero

1. For $i = 1$ to $n$
   1.1. $\text{meanvalue} = 1$; $c = 0$;
   1.2. For $j = 1$ to $n$
      1.2.1. If $a_{ij} \neq 0$ then
         1.2.1.1. $\text{meanvalue} = \text{meanvalue} \times a_{ij}$
         1.2.1.2. $c = c + 1$
      1.2.2. Endif
   1.3. Next $j$
   1.4. $\text{meanvalue} = \text{meanvalue}^{\frac{1}{c-1}}$  
      ‘this is the row’s geometric mean, the $c - 1$ in  
      ‘the denominator is used to compensate for the  
      ‘fact that an element is always equal to itself

1.5. For $j = 1$ to $n$
   1.5.1. If $a_{ij} = 0$ then  
         ‘we assume zero to be a missing value
      1.5.1.1. $a_{ij} = \text{meanvalue}$
      1.5.1.2. $a_{ij} = 1/\text{meanvalue}$
   1.5.2. Endif
   1.6. Next $j$
2. Next $i$

Figure 7 Imputation algorithm.

Because in an ICD with replication factor $r$ the number of judgments made is lower than in the case of the full factorial design we need to change the formula for the Inconsistency Index to reflect the reduced degrees of freedoms. To do this we substitute the denominator in $<6>$ with the number of judgments made $(r \times n / 2)$ minus the number of user stories being estimated $(n - 1)$, which yields:
The computation of the standard deviation of the size of the user stories remains unchanged.

6. Empirical Verification

In this section we compare the performance of the fractional paired comparison method to the results generated by the full factorial method using a set of 15 user stories, see table 3, derived from a popular Canadian job board [17]. The user stories are described in table 3 using the template: “As <role> I would like to <action> so that <benefit>”.

For an evaluation of the paired comparisons method against actuals, refer to [9, 18, 19].

6.1. Method set-up

The verification was conducted using two subsets of different size to explore the impact of the number of user stories in the reliability of the fractional method. Two user stories were included in both sets because we needed 17 user stories, 7 for one subset and 10 for the other, and we only had 15 user stories and we also wanted to use the same story as reference. In table 3, the numbers in the left-hand side column indicates which user stories were included in which dataset (1 or 2, or both).

First a full-factorial estimation was performed on each dataset by a senior software developer. From these two datasets the fractional designs were generated by excluding from the calculations those values that would have been skipped by the ICDs. This approach allowed us to control for the judgment errors associated with repeated questioning.

Figure 8 shows the data and the comparisons included in the first dataset for three different designs with $r = 6$ (full factorial – 100% of the comparisons), $r = 4$ (66% of the comparison) and $r = 2$ (33% of the comparisons). In figure 8, only the shaded values need to be provided by the estimator. The user story ‘Notification’ is the reference user story and its predetermined size is 10 story points.

The full factorial design required 21 comparisons, while the first and second ICDs required 14 and 7 respectively. The results are shown in Table 4.

The second dataset consisted of 10 user stories which were estimated using a full factorial design and 4 different ICDs. Results are shown in Table 5.

Notice that for both datasets, the minimal replication, $r = 2$, ICDs are equivalent to the simple triangulation procedure proposed in the agile literature.
### Table 3 User stories derived from the job board [17]

<table>
<thead>
<tr>
<th>Dataset (1 or 2)</th>
<th>Role</th>
<th>Action</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Login</td>
<td>I can use the system capabilities reserved to registered job seekers</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Logout</td>
<td>To end a session and protect my data from being accessed by unauthorized people</td>
<td></td>
</tr>
<tr>
<td>1,2</td>
<td>Register</td>
<td>I can make my data available to headhunters and use the system capabilities reserved for registered job seekers</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Search job announcements</td>
<td>I can find selected postings based on keywords or criteria such as job category, location, industry and city</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Create career alert</td>
<td>I will get email notifications whenever a new announcement matching the search criteria is first posted</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Suspend career alert</td>
<td>I will not receive further notifications</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Delete career alert</td>
<td>I will not receive further notifications without deleting the career alert</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Upload resume</td>
<td>It can be searched and read by recruiters</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Delete resume</td>
<td>It is not longer available for recruiters</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Login</td>
<td>I can use the system capabilities reserved to registered recruiters</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Logout</td>
<td>To end a session and protect my data from being accessed by unauthorized people</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Post</td>
<td>I can post a new job announcement</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Edit</td>
<td>I can modify an existing job announcement</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Delete</td>
<td>I can delete an existing job announcement</td>
<td></td>
</tr>
<tr>
<td>1,2</td>
<td>System owner</td>
<td>Notify</td>
<td>I can email all jobseekers new postings according to their career alerts</td>
</tr>
</tbody>
</table>
Figure 8 Empirical verification using three different designs for dataset 1

Note 1: Figure 8a) Design with \( r = 6 \) (full factorial – 100% of the comparisons),

Note 2: Figure 8b) Design with \( r = 4 \) (66% of the comparisons)

Note 3: Figure 8c) Design with \( r = 2 \) (33% of the comparisons).
Table 4 Dataset 1: Estimation results of 7 user stories for 3 different replication factors

<table>
<thead>
<tr>
<th>User Story</th>
<th>Estimated Story Points</th>
<th>Std. Dev.</th>
<th>Estimated Story Points</th>
<th>Std. Dev.</th>
<th>Estimated Story Points</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>r = 6 (full factorial)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registration</td>
<td>14.4</td>
<td>0.4</td>
<td>18.8</td>
<td>5.4</td>
<td>31.9</td>
<td>22.6</td>
</tr>
<tr>
<td>Notification (Reference)</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Alert</td>
<td>7.4</td>
<td>0.2</td>
<td>5.7</td>
<td>1.6</td>
<td>6.8</td>
<td>4.8</td>
</tr>
<tr>
<td>Search jobs</td>
<td>5.0</td>
<td>0.1</td>
<td>4.8</td>
<td>1.4</td>
<td>7.5</td>
<td>5.3</td>
</tr>
<tr>
<td>Login job seeker</td>
<td>3.4</td>
<td>0.1</td>
<td>4.5</td>
<td>1.3</td>
<td>9.0</td>
<td>6.4</td>
</tr>
<tr>
<td>Upload resume</td>
<td>1.6</td>
<td>0.0</td>
<td>3.0</td>
<td>0.9</td>
<td>8.6</td>
<td>6.1</td>
</tr>
<tr>
<td>Log out job seeker</td>
<td>1.3</td>
<td>0.0</td>
<td>2.7</td>
<td>0.8</td>
<td>8.0</td>
<td>5.6</td>
</tr>
<tr>
<td><strong>Inconsistency Index</strong></td>
<td>0.06</td>
<td>0.70</td>
<td></td>
<td></td>
<td>1.73</td>
<td></td>
</tr>
</tbody>
</table>

Table 5 Dataset 2: Estimation results of 10 user stories for 5 different replication factors

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>r = 9 (full factorial)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registration</td>
<td>13.0</td>
<td>0.8</td>
<td>13.5</td>
<td>1.6</td>
<td>14.2</td>
<td>2.8</td>
<td>16.4</td>
<td>4.4</td>
<td>29.7</td>
<td>19.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notification (Reference)</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post announcement</td>
<td>6.1</td>
<td>0.4</td>
<td>6.0</td>
<td>0.7</td>
<td>5.2</td>
<td>1.1</td>
<td>4.1</td>
<td>1.1</td>
<td>5.4</td>
<td>3.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit announcement</td>
<td>4.4</td>
<td>0.3</td>
<td>4.3</td>
<td>0.5</td>
<td>3.7</td>
<td>0.7</td>
<td>3.6</td>
<td>1.0</td>
<td>6.7</td>
<td>4.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Login recruiter</td>
<td>3.0</td>
<td>0.2</td>
<td>2.8</td>
<td>0.3</td>
<td>2.7</td>
<td>0.6</td>
<td>3.1</td>
<td>0.8</td>
<td>7.0</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suspend alert</td>
<td>2.3</td>
<td>0.1</td>
<td>2.5</td>
<td>0.3</td>
<td>2.5</td>
<td>0.5</td>
<td>3.3</td>
<td>0.9</td>
<td>7.8</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete alert</td>
<td>1.9</td>
<td>0.1</td>
<td>2.1</td>
<td>0.2</td>
<td>2.7</td>
<td>0.5</td>
<td>3.8</td>
<td>1.0</td>
<td>8.5</td>
<td>5.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete announcement</td>
<td>1.5</td>
<td>0.1</td>
<td>1.7</td>
<td>0.2</td>
<td>2.4</td>
<td>0.5</td>
<td>3.8</td>
<td>1.0</td>
<td>8.4</td>
<td>5.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete resume</td>
<td>1.3</td>
<td>0.1</td>
<td>1.5</td>
<td>0.2</td>
<td>2.2</td>
<td>0.4</td>
<td>3.7</td>
<td>1.0</td>
<td>8.7</td>
<td>5.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logout recruiter</td>
<td>1.0</td>
<td>0.1</td>
<td>1.2</td>
<td>0.1</td>
<td>1.8</td>
<td>0.4</td>
<td>3.1</td>
<td>0.9</td>
<td>7.5</td>
<td>4.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Inconsistency Index</strong></td>
<td>0.18</td>
<td>0.35</td>
<td>0.60</td>
<td>0.81</td>
<td></td>
<td></td>
<td>1.91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.2. Discussion of results

Each ICDs’ performance was evaluated using the Mean Magnitude Relative Error (MMRE) and the Predictive Quality Indicator (Pred) at the user story level and the MRE at the project level. An estimation method is considered acceptable when its MMRE is less or equal than 0.25 and its Pred(.25) is greater or equal than 0.75 [8, 20].
Table 6 present the performance for the estimation results presented in Tables 4 and 5. The values obtained show that the fractional designs produce acceptable results at the project level with very low replication factors \( (r = 4) \). As expected, the influence of inconsistent judgments increases with a reduction in the number of comparisons and this results in higher MMREs and lower Pred(.25)s in the experimental situation.

Notice that as a consequence of retaining the values from the full-factorial design to control for judgment error, we did not correct any values to obtain an inconsistency index closer to the recommended 0.35. Had we allowed ourselves to perform one or two amendments, like we would normally do in practice, would have brought the index down and improved the MMREs and Pred(.25)s in low replication estimations. Reasoning along this line, triangulating against two other user stories \( (r = 2) \) would require almost perfect consistency on the judgments rendered for the extra comparisons to increase the estimate’s reliability.
Table 6 Methods’ performance at the user story and at the project levels

<table>
<thead>
<tr>
<th>Replication Factor (r)</th>
<th>Inconsistency Index</th>
<th>User story level</th>
<th>Project level</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MMRE</td>
<td>Pred(0.25)</td>
<td>MRE</td>
</tr>
<tr>
<td>First data set – 7 user stories</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.70</td>
<td>0.45</td>
<td>0.33</td>
<td>0.14</td>
</tr>
<tr>
<td>2</td>
<td>1.73</td>
<td>2.10</td>
<td>0.17</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Second data set – 10 user stories

|                       |                     |      |            |        |
| 8                     | 0.35                | 0.08 | 1          | 0.02    | Estimates are acceptable at both the project and the user story levels. |
| 6                     | 0.60                | 0.31 | 0.56       | 0.06    | As the inconsistency index raises, individual user stories estimates start to deteriorate. Estimates at the project level remain acceptable. |
| 4                     | 0.81                | 0.78 | 0.33       | 0.22    | Estimates are not acceptable at the user story or the project level. Notice that this is the case of triangulation against 2 other user stories. |
| 2                     | 1.91                | 2.69 | 0.11       | 1.23    | Estimates are not acceptable at the user story or the project level. Notice that this is the case of triangulation against 2 other user stories. |

* The denominator used in both calculations is the number of user stories – 1 to account for the reference element.

\[
\text{MMRE} = \frac{MRE}{(n-1)} = \frac{\sum \text{abs}(x_i - \hat{x}_i)/x_i}{n-1}
\]

\[
\text{Pred}(0.25) = k/(n-1) \text{ is the proportion of observations (k) for which the MRE is lower of equal to 0.25.}
\]

7. Extending the technique to maintenance requests and other fine grained development

Another interesting area of application for the paired comparison method is the estimation of maintenance requests. One could for example batch a number of maintenance requests and then estimate them by comparing them against each other and to a reference request or one could implement a running estimating process where each new request would be estimated against previously implemented requests. See Figure 8.

Comment [L2]: I would leave this out of this paper. To be kept for another paper where 'estimation of effort' is discussed rigorously. A good idea, but not properly positioned.
To implement a running estimation process, one would keep, on a system or product basis, a list with a handful of past requests containing data such as the request’s description and the actual effort required for its implementation. An incoming request would be estimated by comparing it to the existing requests and upon completion added to the list pushing out the oldest request. In this way the effort estimates would reflect the increasing system entropy and the decreasing knowledge that characterize long-lived applications.

Since we are using multiple references to estimate a single maintenance request, the size calculation described in <4> would have to be replaced by <12> which substitutes the single point reference value with the coefficients of a regression line of previously completed maintenance requests.

\[ sp_i = \hat{a} + \hat{b} \times mrs_i \]

\(<12>\)

\(\hat{a}, \hat{b} = \text{regression line coefficients for the } sp_{mr} \text{ used as references}\)

8. Summary

Agile estimation approaches usually start by sizing the user stories to be developed by comparing them among themselves. Different techniques, with varying degrees of formality have been proposed by the Agile community to conduct the comparisons – plain contrasts, triangulation, planning poker and voting. This article adds to these techniques by proposing the use of a modified paired comparison method in which a reduced number of comparisons is selected according to an incomplete cyclical design.

An empirical verification of this proposal was conducted using two datasets showing that the proposed method produces good estimates even when the number of comparisons is reduced by half those required by the original formulation of the paired comparison method. A byproduct of the evaluation is the conclusion that the simple triangulation advocated in the Agile literature does not to automatically result in more reliable estimates.

Those seeking to introduce the method in their organizations must be aware that while people readily buy into the idea of comparing user stories against each other, they tend
to become discouraged by the underlying mathematics. Therefore, two things are required for a successful deployment of the method: first the careful selection of the amount of details to be included in training presentations and process documentation and second, the development of a simple spreadsheet to support these calculations.

9. Acknowledgment

We would like to thank Ahmed Bedhief for patiently going through all the comparisons required by the full factorial experiment.

10. References


