Information and Software Technology 51 (2009) 1327–1337

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Sizing user stories using paired comparisons

Eduardo Miranda [a,*], Pierre Bourque [b], Alain Abran [b]

[a] Institute for Software Research, Carnegie-Mellon University, Pittsburgh, Pennsylvania, PA, United States
[b] École de Technologie Supérieure, Université du Québec, Canada

ARTICLE INFO

Article history:
Received 16 February 2009
Received in revised form 5 April 2009
Accepted 10 April 2009
Available online 18 April 2009

Keywords:
Agile estimation
Triangulation
Estimating by analogy
Paired comparisons
Software sizing
Expert estimation
Expert judgment
Project planning
Incomplete cyclic designs

ABSTRACT

Agile estimation approaches usually start by sizing the user stories to be developed by comparing them to one another. Various techniques, with varying degrees of formality, are used to perform 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 cyclic 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.

1. Introduction

Agile estimation approaches typically comprise three steps: (1) comparison of the user stories to be developed to one another for the purpose of establishing their relative size; (2) conversion of the size estimates to lead times using an assumed team productivity; and (3) re-estimation of the project lead times using the team's actual productivity, once this becomes known after two or three iterations.

User story comparisons take the following form: "This story is like that story, so its size must be roughly the same," or "This story is a little bit bigger than that story which was estimated at 4, so its size should be around 5." The numbers 4 and 5 in the previous sentence are called "story points", which are numbers in ratio scale purportedly proportional to the effort it would take to develop each story based on its perceived size and complexity [1]. 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, to triangulation – the comparison of a user story with two others, to a number of Delphi [2] like techniques such as the planning poker [3]. 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 to be of the same size [4].

The project lead time is calculated using the concept of *velocity*, which is a proxy for the productivity of the team. At first, 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.

As will be shown later, comparing one user story to another, or to two others, is not good enough to produce reliable estimates. The first reaction to this is to increase the number of comparisons, but this creates some problems of its own. As even the most devoted estimator gets tired after making a large number of comparisons, the question of how many comparisons to make becomes really important, as does the problem of dealing with the inconsistencies inherent to the judging process.

To address these problems, we propose the use of incomplete cyclic designs to identify which user stories to compare with which to reach a desired accuracy, and the use of the paired comparison method [5–7] to deal with judgment inconsistencies.

The rest of the paper is organized as follows: Section 2 formalizes the triangulation concept, Section 3 explains the basic paired comparison method, Section 4 presents the modified process using incomplete cyclic designs, Section 5 discusses the accuracy and

* Corresponding author.
E-mail address: mirandae@andrew.cmu.edu (E. Miranda).

0950-5849/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2009.04.003

precision of the resulting estimates, and Section 6 provides a summary of the article.

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 reliability[1] of the estimate [3]. When using triangulation, the comparisons sound something like this: "I'm giving user story B 2 points, because it feels like its implementation will take somewhat more effort than user story A, which I already rated at 1 story point, and somewhat less effort than user story C, which I rated as a 4-point story." Despite its intuitive appeal, triangulation is not as simple as the sentence above makes it appear. First, there is the problem of consistency, which can be mathematically expressed as:

$$a_{ij} \times a_{jk} = a_{ik} \;\forall i,j,k \in n \quad (1)$$

Eq. (1) reads as follows: if user story$_i$ is a_{ij} times bigger[2] 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 by arranging the user stories in a circular pattern and linking those being compared (see Fig. 1). Given n user stories to be estimated, there are $n(n-1)(n-2)/2$ possible configurations or designs which can be evaluated, but not all are equally good. A good design must have two properties: balance and connectedness [8–10]. A design is considered balanced when every user story appears in as many comparisons as any other user story. This ensures that one user story does not overly influence 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. Fig. 1b 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. In all the designs shown in Fig. 1, r is 2.

Balance and connectedness are necessary, but not sufficient conditions for a good estimation. As shown by Burton [8], a low r, such as that used in the triangulation approach ($r = 2$) is very sensitive to errors in judgment, and thus tends to produce unreliable results. In his experiments, Burton found that the correlation (ρ) 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, this is discussed later.

3. Paired comparison method basics

3.1. 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. After this is done, one of the n user stories is assigned an arbitrary number of story points. Using this story as reference, the sizes in story points, of all the other user stories are calculated. The process is called Full Factorial Pairwise Comparison because it compares all user stories (factors) against one another, see Fig. 2.

Although the selection of the user story to be used as reference and the allocation of story points to it is arbitrary to a certain point,[3] a consistent selection and allocation, i.e. two comparable user stories are not allocated 4 story points in one project and 10 in other, is useful for the developers to develop an intuition or sense for the effort required in the realization of a user story with so many story points.

It is also possible to use the method to estimate the effort required by each user story instead of their story points. In this case, a user story whose development effort, from either a previous project or a *spike*,[4] is known will be brought in as reference story. For a more detailed description of the method, refer to [5,6].

In the rest of the document we will work with story points to remain true to the title of the essay but all the same concepts apply to the calculations using effort.

3.2. The pairwise comparison of user stories

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

$$A^{nxn} = \begin{cases} a_{ij} = \frac{sp_i}{sp_j} & \text{How much bigger(smaller)user story}_i \\ & \text{is with respect to user story}_j \\ a_{ii} = 1 & \text{Every user story has the same size as itself} \\ a_{ji} = \frac{1}{a_{ij}} & \text{If user story}_i \text{ is } a_{ij} \text{ times bigger (smaller)} \\ & \text{than user story}_j\text{, then user story}_j \\ & \text{is } 1/a_{ij} \text{ times smaller(bigger)than user story}_i \end{cases} \quad (2)$$

sp_i and sp_j are the as yet unknown numbers of story points for user story$_i$ and user story$_j$ to be derived from the a_{ij} judgments. Note 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}.

3.3. Calculating the size, the Inconsistency Index, and the standard deviation

Once all the a_{ij} judgments have been recorded in the judgment matrix, the mean relative size (mrs_i) of user story$_i$ is calculated as the geometric mean [11,12] of the ith 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 [5,6].

$$mrs_i = \left(\prod_{j=1}^{n} a_{ij}\right)^{\frac{1}{n}} \quad (3)$$

$$sp_i = \frac{sp_{reference}}{mrs_{reference}} \times mrs_i \quad (4)$$

As inconsistencies are inherent to the judgment process, Crawford and Williams [12] and Aguaron and Moreno-Jimenez [13] suggest

[1] A reliable sizing method will yield estimates that are accurate, that is, close to their true value, and precise, that is estimates must be consistent across repeated observations in the same circumstances.

[2] The comparison can go both ways, i.e. replacing bigger for smaller.

[3] The number zero must be reserved for "stories" with not content to preserve the properties of a ratio scale.

[4] In the Agile terminology a spike is an experiment that is performed to learn something. In this case the spike would consist on developing a user story tracking how much effort it required.