Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate

ACCURACY OF EFFORT ESTIMATION MODELS: EVIDENCE OF BASELINE EFFECTIVENESS

This briefing reports scientific evidence on the effectiveness of both new and previously defined baseline methods for evaluating the accuracy of effort estimation methods.

FINDINGS

There is no doubt that it is very important to software companies to estimate software development effort accurately. So, *it is important to assess the accuracy of effort estimation models*.

We propose a new, general framework for evaluating effort estimation models.

The general idea is to evaluate the accuracy of an effort estimation model against a reference model.

- If the accuracy of the model is greater than the expected accuracy of the reference model, the model is a candidate for practical use;
- Otherwise, the model is not sufficiently accurate.

So, *what is an adequate reference model*?

Previous work suggested the use of random reference models. The random reference model estimates the effort of a new project by randomly picking the value of the effort of a project in a pool of previous project.

Clearly, if an effort estimation model is less accurate than what can be expected of random guessing, the model is not accurate enough.

Also, we want the effort estimation model to be not just better, but *significantly better* than the random reference model. We may want to use an effort estimation model that is better than random guessing 95% of the time.

We believe, however, that the bar should be set higher. We can build better reference models with the same information used for random guessing. For instance, even a *constant model* where effort is estimated as the simple mean of the effort of previous projects would be a *more accurate estimation model than random guessing*. So, we propose that a software manager should obtain the basic estimate using the constant model, instead of random guessing.

More accurate models can be built by correlating some software characteristics to effort. How much more accurate are predictions based on models where these characteristics are the independent variables than predictions based on the simple mean?

Our framework

- Is *general* enough to be used with *any cost function*;
- Uses as *reference model the best possible constant model* for the chosen cost function;
- Is based on the *comparison of the accuracies of the model and the reference model*;
- Checks whether the *model is also significantly better than the random reference model*.

We introduced a new indicator of model accuracy. *IARA (Individual Absolute Residual Accuracy)* is the probability that the effort estimation model provides better results than the random model in the majority of estimates it gives.

An extensive data analysis shows that

- *The constant model sets a higher bar than the random model;*
- *The constant model is approximately better than random guessing approximately 95% of the time, but much easier to compute;*
- *IARA appears to be a good indicator of the accuracy of estimation models.*

Keywords

Effort estimation models, Estimation accuracy, Model evaluation

Who is this briefing for?

Software engineering practitioners who want to evaluate the accuracy of estimates obtained via effort estimation methods, based on scientific evidence.

Where do the findings come from?

All findings of this briefing were extracted from the empirical study conducted by Lavazza and Morasca (http://dl.acm.org/citation.cfm?id=3084226&picked=prox).

What is included in this briefing?

The main findings of the original study.

What is not included in this briefing?

Detailed description of all the findings.

To access other evidence briefings on software engineering:

http://ease2017.bth.se/

For additional information about our research, please contact

luigi.lavazza@uninsubria.it
sandro.morasca@uninsubria.it

ORIGINAL RESEARCH REFERENCE

Luigi Lavazza and Sandro Morasca ***On the Evaluation of Effort Estimation Models***. *21st International Conference on Evaluation and Assessment in Software Engineering,* 2017.