

Understanding the Human Estimator

Gary D. Boetticher, Nazim Lokhandwala, James C. Helm

University of Houston – Clear Lake

Boetticher@uhcl.edu, Nazim.Lokhandwala@sprint.com, Helm@uhcl.edu

Abstract

Among the various forms of estimation (human-based, algorithmic, and machine learners), human-based remains the predominant methodology of choice [1]. Algorithmic-based (e.g. COCOMO, FPA) approaches rely heavily upon human intervention for supplying estimates for many of the sub-components. Understanding the role of the human estimator is critical for improving the effort estimation process. Every human estimator draws upon his or her background of domain knowledge, technical knowledge, experience, and education in formulating an estimate. This research uses estimator demographic information and constructs various statistical and machine learner models with a best case result of 93% accuracy. Furthermore, additional experiments break the demographics into major categories, education, work, and domain experience in order to gain greater insight into the estimator. The resulting models are beneficial in predicting the reliability of the estimator.

1. Introduction

Rather than build more predictor models based on effort estimates, or construct/refine current algorithmic models, *Why don't empirical software engineer researchers focus on the humans making the estimates?*

Of all the effort estimation techniques available, human-based estimation remains the most popular due to its simplicity and flexibility in estimating input and time spent on producing estimates. Various studies compiled by Jorgenson [1], show that human-based estimation is the preferred technique over algorithmic and/or machine-learning approaches about 77.4% of the time. Furthermore, substantial evidence does not exist that supports any other method, which gives guaranteed better estimates than human-based estimation [2].

Algorithmic-based estimation approaches are based on human subjectivity. The post-architecture intermediate COCOMO model 23 parameters (e.g. scale factors and effort multipliers) that require the modeler to discriminate between classes and to weigh/consolidate different sub-terms within one parameter. For example, the *Process Maturity* scale factor is a consolidation of 18 key process

areas. Finally, the COCOMO model relies upon an accurate *size* estimate. This size metric may be based on a *source lines of code* (SLOC) estimate, or it may use Function Points. Deriving Function Points requires the user to assign values for the 14 Global System Characteristics.

Defining all the factors and coming up with the estimate using the model does not conclude the estimating effort. The estimator must calibrate the results from estimating models to current projects and organization environments in order to achieve potentially accurate results [3]. Even in Function Point Analysis there exists an imperative need of the subjective estimator's judgement in rating the General System Characteristics. Thus, algorithmic techniques depend heavily upon human, preferably expert, intervention. It seems that human-based estimation is unavoidable.

Machine Learning (ML) based estimation requires many human decisions in terms of which metrics to collect, number of samples to collect, which learner to apply, and how to interpret the results. The black box nature of some ML, such as neural networks, introduces an additional learning curve that might discourage estimators from using it, until they have successfully tried it several times in order to build their confidence in it.

A challenge that exists in human estimation revolves around nature of human life. Assume a person starts working in the software field at the age of 24 and retires at the age of 65. The typical length of a software project is two years. On average, a person would encounter approximately 21 projects during their career. This implies that there are relatively few benchmark points on which to base current estimates. Furthermore, as software development increases in complexity and spans over more complex and dynamic domains, it becomes harder to apply historical domain knowledge in the current domains with newer technologies.

There has been a continuous effort to enhance algorithmic models by calibrating them in order to measure the impact of various inputs on the accuracy of outputs received from the algorithmic models [4]. However, no such work exists in expert estimation on how to estimate more efficiently. There does not appear to be any research which assesses human characteristics such as age, gender, years of experience, domain experience, etc. as a basis for predicting a person's ability to estimate.

This research examines the influence of different human demographics on the estimation process to determine the impact of demographics in the estimation process. There are significant reasons for addressing this topic. The knowledge of the programmers' demographics can be used by project managers, in the context of bottom-up estimation, to calibrate a programmer's estimates based upon their demographics. Similarly, in a Delphi group scenario, several estimators might produce different estimates. These different estimates may be reconciled by assigning weights to the estimates based on their individual demographics. At an individual level, those affiliated with project estimation can concentrate on those factors that have maximum impact on estimation and work towards their self-development in order to improve their estimation accuracy and improve their Personal Software Process.

To assess human estimators, a survey is developed which gathers user demographic and requests the respondent to estimate the time needed to complete 28 separate components. Results from 122 different respondents are analyzed using various machine learning and statistical techniques.

2. Related Research

There have been a relatively few studies on expert estimation. Gray [5] examines a set of expert-derived estimates for the effort required to develop a collection of modules from a large health-care system. Statistical tests suggest a clear relationship between the type (screen or report) and characteristics (size and type of modules subject to changes) of modules and the likelihood of the associated development effort being underestimated, approximately correct, or over-estimated.

Connolly [6] compares *Decomposed* versus *Holistic* Estimates of Effort Required for Software Writing Tasks. He reports that the actual effort used to solve programming tasks falls inside the 98% confidence effort prediction intervals for only 60% of the tasks. Explicit attention to and training in establishing good minimum and maximum effort values increases the proportion inside the prediction interval to about 70%. He suggests that expert estimates get more accurate when including risk analysis in the estimation process.

Jorgensen [7] randomly selected 109 maintenance tasks and assigns them to people with varying experiences after providing details regarding task specifications. The study reports no clear correlation between length of experience and prediction accuracy of own work among software maintainers.

Studies from other domains indicate several interesting characteristics of expert judgement that can probably be relevant to software effort estimation. Hoch [8] in his study on decision support systems suggests that experts

performed better than models in a highly predictable environment, but worse in a less predictive environment. MacGregor's [9] study on aids for quantitative estimation suggests that decomposition of a task for estimation purposes could activate too much information processing and lead the expert estimator astray. Braun [10] compares expert judgment with model forecasts suggests that experts outperformed models in shorter-term business forecasting, whereas models outperformed experts in long term forecasting. The application and relation of these characteristics in software effort estimation have not been found.

3. Survey-Based Data

A Web-based survey serves as the data collection mechanism. This survey consists of three sections. The first section gathers demographic information about the respondent. In the second section, the respondent assesses the amount of effort, in hours, for a set of 28 modules. The third section provides statistical feedback to the respondent. This survey is available at:

<http://nas.cl.uh.edu/boetticher/EffortEstimationSurvey.html>

3.1 Section 1: Demographic Information

Participant demographics include: *Year Of Birth*, *Gender*, *Nationality*, *Highest Academic Degree Achieved*, *The Number of Courses taken at the Undergraduate and Graduate level* (Computer Science/Computer Information Systems, Computer Hardware, Management Information Systems, etc.), *The Number of Workshops and Conference attended based on content* (Computer Science/Computer Information Systems, Computer Hardware, Management Information Systems, Project Management or Project Metrics, Software Engineering), *Number of Years of Industrial Experience* in a specific programming language (18 choices), *Years of Work Experience* in Hardware and Software Industry, *Years of Experience as a Project Manager* in Hardware and Software Industry, *Number of Projects* estimated in Hardware and Software Industry, and *Average Size of Software Projects* estimated.

3.2 Section 2: Component Assessment

In the second phase the respondent must provide effort estimates for a set of 28 different modules. These modules originated in an eCommerce project developed by one of the authors (Boetticher) in the late 1990s. Rigorous effort estimation data was logged per module during the development process. To insure the survey could be completed within a reasonable amount of time, a

representative sample of modules from the project are included in the survey.

The survey provides extensive help in the form of an overall description of the whole project along with context sensitive help per module. Figure 1 illustrates one of the modules from the survey.

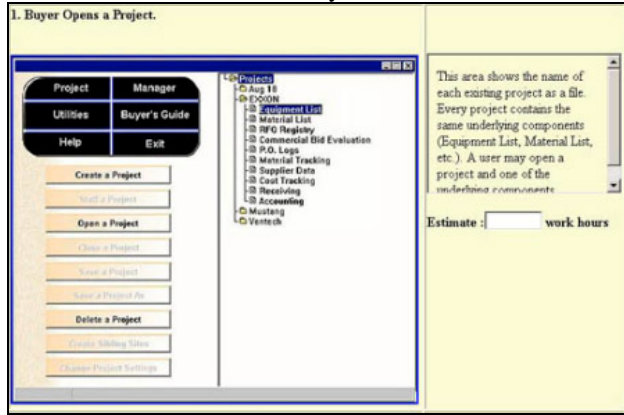


Figure 1: Screenshot of one of the modules

This section closes with questions regarding the respondent's domain experience in the procurement and process industry.

3.3 Section 3: Survey Results

After assessing the 28 modules, the respondent receives feedback regarding their estimates. Figure 2 shows a graph from a survey where the results are sorted by effort. Ideally, a respondent's estimates would overlap the actual values. This graph also provides a Pred(25) count, which represents the number of estimates within 25% of the actual values.

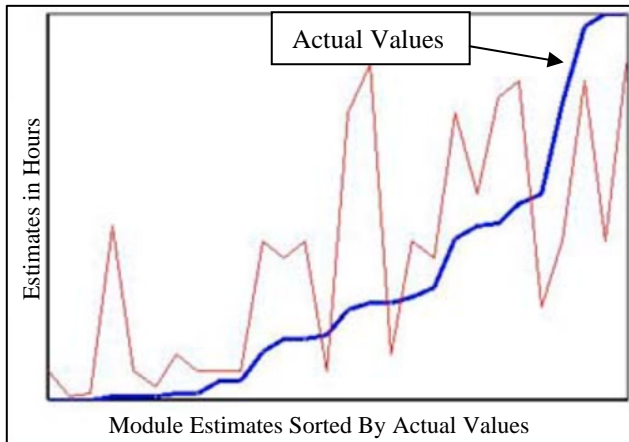


Figure 2. Estimates Sorted by Actual Values

The survey also provides project-based feedback to the respondent in terms of how their accumulative estimates compare to previous participants. Figure 3 shows the Mean Absolute Relative Error (MARE) of all the

participants plotted in ascending order. In this example, the participant was in the top 80% profile.

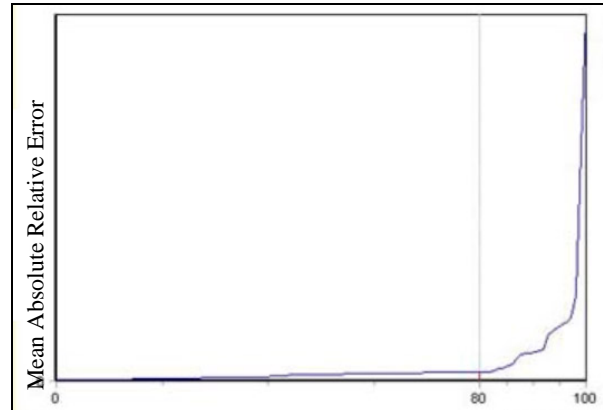


Figure 3. Respondent's Estimates Relative to Other Participants in terms of MARE.

The module and project feedback offer immense value to the respondent and is intended to motivate the user to complete the survey.

4. Data Demographics

The data set consists of 122 samples that were collected from 2001 through 2004. The average age is 29.67. There are 100/22 male/female respondents. Academically, in terms of highest degree held, 23% held a Master's and 74% held a Bachelor's degree. Citizens from 20 different countries completed the survey with 48% from India, 24% from the United States, and 8% from Romania. Table 1 summarizes each participant's work, estimation, and domain experience.

Table 1: Summary of Experience of Participants

	Ave. Years	Max. Years	Std. Dev.
Years of Experience as a Hardware Proj. Mgr.	0.6557	15	1.9251
Software Proj. Mgr.	1.3443	10	2.0811
No. of Projects estimated			
Hardware Projects	0.8279	20	2.6307
Soft. Projects	2.9508	28	4.4848
Domain Experience			
Procurement & Billing	0.6209	10	1.3818
Process Industry	0.7274	20	2.2512

5. Experiments

The first set of experiments seek to determine the contribution of specific demographic features in formulating accurate estimates in terms of Magnitude of

Relative Error (MRE). All demographic features are analyzed followed by a study which focuses on education, work experience, and domain experience respectively.

Data is assessed using non-linear statistical models and genetic programs (GP). One reason for selecting these approaches is that both techniques provide human readable solutions.

Using all the demographic attributes, five progressively complex non-linear models are constructed. Table 2 shows the results from these models.

Table 2: Non-Linear Regression Impact (All Factors) on Estimation MRE

Type Of Regression	R ²	Standard Error
Exponential Regression	0.8847	1.6470
Second Order Polynomial	0.3656	4.1435
Third Order Polynomial	0.5811	3.6531
Fourth Order Polynomial	0.8014	2.7730
Fifth Order Polynomial	0.9294	1.8682

The exponential regression and the fifth order polynomial produce the best results (93% R²) and show that it is possible to identify a pattern using all the demographic factors. Equation 1 shows the actual equation for the exponential regression model.

$$\begin{aligned}
 EstimationMRE = & \exp(-9.4450 * Deg + 0.2772 * \\
 & TechUGCourses + 0.6474 * TechGCourses - 4.3233 * \\
 & MgmtUGCourses - 0.7721 * MgmtGCourses - 1.0025 * \\
 & TotWShops - 0.0472 * TotConf + 0.5699 * TotLangExp + \\
 & 3.4556 * HWPMEExp + 0.4810 * SWPMEExp - 2.0149 * \\
 & HWProjEstExp - 0.1960 * SWProjEstExp + 1.6498 * DomExp \\
 & + 1.5815 * ProcIndExp + 9.0745) \quad (Eq. 1)
 \end{aligned}$$

For the GP, a series of subexperiments are conducted which vary the GP settings in terms of chromosome length (how many characters may be in an equation) and the number of generations. Twenty trials are conducted for each configuration. Table 3 shows the best results after performing 20 trials for each GP configuration.

Table 3: GP Results for All Factors on MRE

Chrom. Length	Gen.	R ²	Stand. Error
1000	50	0.7739	2.2952
512	128	0.7726	2.3021
1000	128	0.7043	2.6250

The equation for the best GP model is:

$$\begin{aligned}
 & ((MgmtGCourses \wedge (((Log (((TotLangExp / (TotLangExp / \\
 & TechGCourses * HWPMEExp))) - (TechGCourses * \\
 & HWPMEExp)) - ((Sin (MgmtGCourses \wedge (Sin ((TechGCourses * \\
 & HWPMEExp) - (MgmtGCourses \wedge (((Log (HWPMEExp \wedge \\
 & (TotLangExp / (TechGCourses * HWPMEExp))) - (Abs (Log \\
 & ((TotLangExp / (TechGCourses * HWPMEExp)) - ((Sin (Abs
 \end{aligned}$$

$$\begin{aligned}
 & (TechUGCourses / MgmtGCourses))) - (TotLangExp / \\
 & (MgmtGCourses \wedge (((Log (((TotLangExp / (HWPMEExp / \\
 & SWProjEstExp)) - (Sin (TotLangExp / (TotLangExp / \\
 & ((MgmtGCourses \wedge (Log (TechGCourses * HWPMEExp)) - (Sin \\
 & (Abs (Log ((HWPMEExp / SWProjEstExp) - (TechGCourses * \\
 & HWPMEExp)))))) + ((Sin (TechGCourses * HWPMEExp)) - (Sin \\
 & (TechUGCourses / MgmtGCourses)))))) - (Sin \\
 & (TechUGCourses / MgmtGCourses))) - (TechGCourses * \\
 & HWPMEExp)) - (Sin (TechUGCourses / MgmtGCourses)))))) - \\
 & (HWPMEExp / SWProjEstExp)))) - (Sin (TechUGCourses / \\
 & MgmtGCourses)))))) - ((Sin (Abs (Log ((TotLangExp / \\
 & (TechGCourses * HWPMEExp)) - ((Sin ((Sin (Abs (Log \\
 & (HWPMEExp \wedge (TotLangExp / (TechGCourses * HWPMEExp)))))) \\
 & - (TechGCourses * HWPMEExp)) - (HWPMEExp / \\
 & SWProjEstExp)))))) - (Sin (TechUGCourses / \\
 & MgmtGCourses)))))) - (TotLangExp / (TechGCourses * \\
 & HWPMEExp)) - (Sin (TechUGCourses / MgmtGCourses))) + \\
 & (TotLangExp / (TechGCourses * HWPMEExp)) \quad (Eq. 2)
 \end{aligned}$$

where

- Deg* is Numeric Highest Degree Earned,
- TechUGCourses* is Technical Undergraduate Courses,
- TechGCourses* is Technical Graduate Courses,
- MgmtUGCourses* is Managment Undergraduate Courses,
- MgmtGCourses* is Management Graduate Courses,
- TotWShops* is Total Workshops attended,
- TotConf* is Total Conferences attended,
- TotLangExp* is Total Language Experience,
- HWPMEExp* is Hardware Proj. Management Experience,
- SWPMEExp* is Software Proj. Management Experience,
- HWProjEstExp* is Hardware Proj. Estimation Exp,
- SWProjEstExp* is Software Proj. Estimation Experience,
- DomExp* is Domain Experience, and
- ProcIndExp* is Process Industry Experience.

The next set of experiments extract out specific demographic attributes, in terms of education, work experience, and domain experience to determine whether a particular set of attributes is primarily responsible for one's ability to accurately predict effort. Table 4 shows the average of the results of applying non-linear regression and various GP configurations (20 trial for each GP configuration).

Table 4: R² Values of Various Factors on Estimation MRE

	Non-Linear Regress	Genetic Programming Average of 20 Trials			GP
		1000 Chrom 50 Gen	512 Chrom 128 Gen	1000 Chrom 128 Gen	Best Results
Education	0.2136	0.1406	0.1361	0.1973	0.2784
Work Exp.	0.3698	0.5290	0.5328	0.5564	0.7572
Domain Exp.	0.3260	0.5337	0.5405	0.5458	0.5911

On average, the results are fair. The work and domain experience results are comparable while the educational results are low. Considering that 97% of the respondents have either a Bachelor's or Master's degree, the

educational experiments have only two-instance values for assessing effort prediction. Thus, making it difficult to build a good GP model.

6. Discussion

The equations presented in the previous section provide a methodology for assessing the reliability of an estimator. Thus, if the demographics of an estimator are known, it is possible to predict how well they will estimate a project in terms of MRE.

It is interesting that the factor analysis presented in Table 4 shows a very low correlation between education in estimation accuracy. Equation 1 supports this analysis where " $\exp(-9.4450 * Deg.)$ " indicates that the degree status drives towards a zero value irrespective of the number of degrees attained.

There are several challenges in building an accurate model for assessing the estimator. In the proceeding experiments, the actual estimates from the survey are based on the developmental efforts of one person. Thus, a respondent estimates his/her ability to code the project which may not conform to the actual estimates. Also, the underlying project for the survey was written in Delphi. If a participant is not familiar with Delphi, he/she may not estimate well due to no prior Delphi experience. Finally, there is a challenge in terms of the implied historical context associated with the survey's project. New development tools and technologies will skew the estimates associated with the project. Thus, the survey's utility may decay over time.

Software reuse may be perceived as an additional issue in terms of how to amortize reuse efforts within a project. This issue disappears when examining project, as opposed to component, estimates.

Design with reuse at the project perspective (reuse spanning multiple projects) does impact project estimation. This issue warrants further research.

7. Conclusions

Traditionally, issues like model-based versus expert-based effort estimation have served to divide the software engineering discipline into two camps (expert-based, model-based) which, for the most part, ignore each other. This paper is an excellent example of why this division is artificial and should be rejected.

A key result is that expert-based methods can still be rigorously analyzed by model-based methods; i.e. information systems need to know more about computer science (the reverse might be true, but that is beyond the scope of this paper).

8. Future Directions

Several experiments were conducted to reduce model complexity including TAR3 and attribute reduction. These did not produce fruitful results. As more data is collected it is anticipated that simplified models that produce superior results could be generated.

9. References

- [1] Jorgensen, M., "A review of studies on Expert Estimation of Software Development Effort," *Journal of Systems and Software*, 2004.
- [2] Jorgensen, M., "Top-down and Bottom-Up Expert Estimation of Software Development Effort," *Journal of Information and Software Technology*, 2004.
- [3] Jeffery, D. G. and G. Low, "Calibrating estimation tools for software development," *Software Engineering Journal*, vol. 5, no 4, Pp. 215..221.
- [4] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R. and R. Selby, "Cost models for future software life cycle processes: COCOMO 2.0," *Annals of Software Engineering*, Vol. 1, 1995, Pp. 57..94.
- [5] Gray, A. R., MacDonell, S. G. and M. J. Shepperd, "Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgement," *Proceedings of the Sixth International Software Metrics Symposium*, 1999.
- [6] Connolly, T., and D. Dean, "Decomposed versus holistic estimates of effort required for software writing tasks," *Management Science*, vol. 43, 1997, Pp. 1029..1045.
- [7] Jorgensen, M., Sjoberg, D. and G. Kirkeboen, "The Prediction Ability of Experienced Software Maintainers," *4th European Conference on Software Maintenance and Reengineering*, Zurich, 2000.
- [8] Hoch, S. J. and D.A. Schkade, "A psychological approach to decision support systems," *Management Science*, vol. 42, 1996, Pp. 51..64.
- [9] MacGregor, D. G., and S. Lichtenstein, "Problem structuring aids for quantitative estimation," *Journal of behavioral decision making*, vol. 4, 1991, Pp. 101..116.
- [10] Braun, P.A., and I. Yaniv, "A case study of expert judgement: Economists' probabilities versus base rate model forecasts," *Journal of Behavioral Decision Making*, vol. 5, 1992 Pp. 217..231.