Crystal Method For Accurate Software Duration Estimation

Periasamy Sobah, FTMK, and Nanna Suryana Herman, FTMK

Abstract-**All projects share one common characteristic that is the projection of ideas and activities into new endeavours. The ever-present element of risk and uncertainty to the events and tasks leading to completion can never be foretold with absolute accuracy. The software projects are different from other projects. Underestimation is the root course of many software projects not being able to meet the deadline, or failure. Some of the reasons for inaccurate estimation are as follows: the traditional model not being able to capture the project in detail, quick and reliable strategic analysis. The influence of human factor is not able to incorporate explicitly. Failure to consider rework phenomenon. Failure to capture dynamic interaction between technical development and management policies.**

Index Term - **Duration estimation error, Drawback of traditional estimation, Gamma for duration estimation, Software development.**

I. INTRODUCTION

The purpose of project management is to foresee or predict as many of the dangers and problems as possible and to plan, organize and control activities so that projects are completed as successfully as possible in spite of all risks.

PERT (Project Evaluation Review Technique)/CPM (Critical Path Method) is a technique that can be used to plan, schedule and control activities that must be completed to finish a project. But PERT/CPM usually fail to provide an accurate estimate for large-scale project completion time even it is stated that planned project schedules, obtained using critical path analysis, would be optimistic. Project management scholars and practitioners accept that the PERT/CPM fails to provide an accurate estimation for large-scale project completion time [11]. The PERT formula to calculate expected (mean) activity duration times, which are supposed to follow beta density functions, considers three parameters (minimum, most likely, and maximum), when in fact the beta distribution has four parameters (two range parameters and two shape parameters). It turns out that the PERT formula used to calculate the mean as a function of the minimum, most likely, and maximum activity duration time estimates, ignores how the biases to the right or left (related to the variance) affect the shape of the beta distribution. This technique gives an average amount of three variables, which unable to accept.

One of the main factors affecting any product development schedule is rework [2]. The rework done in each stage of software development, is usually double the cost and project duration. Rework delay is usually omitted in the estimation process, which leads to underestimation. Software estimation, where volume (size, cost, effort, resources, risk, skill required) and duration, is not directly proportionally related. Therefore software estimation is becoming a challenge for the project managers to estimate size, cost, effort, resources, risk, skill required to complete a task. The factors affect software project estimation shown in Fig. 1.

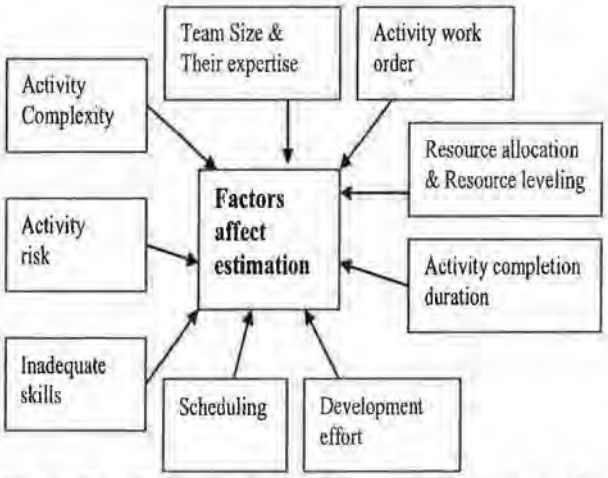

Fig 1: The factors affect software project estimation.

Any improvement in the accuracy of prediction of development effort, duration, and resource requirement can significantly reduce the costs from inaccurate estimation, misleading tendering bids and disabling the monitoring progress. Accurate modeling can also assist in scheduling resources and evaluating risk factors.

The next section presents background covering literature review and industry practices in the software estimation. Section 3 elaborates Gamma distribution for better estimation. Section 4 The EnhnacedEste estimation framework. The conclusion, proposed work & future direction is discussed in section 5.

II. ESTIMATION CHALLENGES IN SOFTWARE PROJECTS

Project managers are assisted by estimation techniques such as Work Breakdown Structures (WBS), Function Point (FP) Analysis, Delphi methods and COCOMO method. Most of the time, these methods are extremely difficult to use and confusing to interpret. They often require extensive amounts of data (such as the number of lines of code, or detailed system design documents) before producing even the most general estimates. In addition, these methods are so focused on being meticulous and precise that they often overlook the obvious advantage of intuitive reasoning. By being consumed in numbers and calculations (or boxes and charts, in the WBS technique), the obvious clues from the available documents are usually missed. Under estimation a project leads to under staffing it (resulting in staff burnout), under-scoping the quality assurance effort (running the risk of low quality deliverables), and setting too short a schedule [17].

A. Reasons for Inaccurate Duration Estimation

The prime reasons for project failure are poor estimation of effort and schedule [18]. But, project duration estimation and resource allocation are treated as two separate problems. This approach is tedious and inadequate for large-scale software project. The project duration estimation and resource allocation need to be combined together to generate near-optimal project estimation. Each project activity is associated with a set of constraints, which specifies the requirement for completing the activity. These constrains can be future classified as resource constraints (specify what kind of resource required) and complexity constraint (describe how much effort is needed for that activity) where must not ignore during estimation. This is especially an important highlight for the study.

The idea using past effort data recorded for completed project tasks is to predict the effort needed for subsequent activities. A report stated standard proportions of effort for particular development activities does not apply. Estimating effort on standard proportions basis would not have improved the management of the projects, would be another reason for wrong estimation [14].

Parametric models develop estimates through mathematical formulas that often use statistical relationships between the size and software characteristics that affect size (e.g., programming language). Some software requires many lines of code whereas others require a few lines of code only. This is important evidence why parametric estimation cannot be used [15].

The well known effect of inaccurate software estimation is schedule overruns. Software estimation errors generally result from four major risk areas. The first is inability to estimate the size of the software project. The inability to accurately specify a development environment that reflects reality would be a second risk. Followed by improper assessment of staff skills, then lack of well-defined objectives, requirements, and specifications during the software development life cycle [4]. A review of surveys indicated the average effort overrun of software projects seems to be in the range 30 to 40 percent, i.e., the average estimation error of software projects is high [13].

B. Duration Estimation and Resource Allocation

The work breakdown structure (WBS) is the key part of the traditional project work plan. WBS is the most common method to allocate resources for a particular task. A very small project presents no great difficulty to allocated resource and adjusts previously assigned resource. But projects with a hundred different tasks cannot be planned with smooth resource allocation. The biggest problem with any manual charting method, however, is too inflexible. A change of plan to any, except very tiniest, project can result in hours of tedious work in

repositioning all the tasks on the chart. This is always coupled with the risk of introducing logical or other errors. Of course, most projects have more than a hundred activities, and manual resource allocation becomes even more difficult (impossible altogether for really large projects). The task risk and effort required to complete are not considered [5]. WBS provides a hierarchical view for the whole project, but the precedence relationships among the work packages are not clearly identified in the WBS [9]. The engineer's expertise is not evaluated here contributing for difficulties for resource allocation and reallocation.

C. *Estimation and Selecting Right Mix of People*

Today's software requirements are complex, and require the united skills of members. Modern organizations are formed based on the limits and constraints of skills and mental potential of individuals [10].Project manager need good skillful people to be placed on a particular task must perform their work keeping in view the time and quality output. Hence, it is not the number of people that serves the purpose; it is the skill level of the people that fulfils the effort to be put on the task. Wrong staff placement for project activity will create room for risk throughout the project.

Some factors that to be need considered during estimate are team's capability by the qualification and experience and the software development environment. The factors such as complexity of the software, required reliability, size of data base, required efficiency (memory and execution time), analyst and programmer capability, experience of team in the application area, experience of team with the programming language and computer, use of tools and software engineering practices, required schedule, and team selection and acquisition highly influence project completion.

Selecting the right mix of people, with both technical and non-technical skills, is a decision that can influence the outcome of project. Although a project manager should strive to acquire the brightest and the best, project team members should be chosen based on the following skills. Technology skills depending upon the nature of the project, members with specific technology skills set as programmers, system analysts, network specialists, and so forth will be required. Business/Organization skills include knowledge or expertise within a specific domain (e.g. compensation planning) as well as knowledge of a particular organization or industry to augment the technical skill requirement. Interpersonal skills are important not only for the team members to understand one another, but also for nature of many projects, others desirable characteristics should include creativity, a tolerance for ambiguity, acceptance of diversity, flexibility in adapting to different roles, and the capacity to take calculated risks.

Skill level is the expertise the project member should have to perform the task given to that member. The skill level of a member is influenced by the experience, behavioral aspects like commitment, and attitude [12]. An investigation of the relationship between experience level and effort estimation accuracy indicates that more experienced engineers more often finish their work packages before the initial estimate.

III. GAMMA DISTRIBUTION AND ENCHANEDESTE FOR PROJECT ESTIMATION

A Gamma distribution is a general type of statistical distribution that is related to the Beta distribution and arises naturally in processes for which time to complete a task between Poisson distributed events are relevant. The gamma function belongs to the category of the special transcendental functions. The gamma distribution includes the chi-squared, Erlang, and exponential distribution as special cases, but the shape parameter of the gamma is not confined to integer values. The gamma distribution starts at the origin and has a flexible shape [3].

The Gamma distribution has several important properties. First, the shape is controlled by two independent non-negative parameters. The values for these two parameters strongly affect the shape of the resulting probability density function (PDF). Second, the PDF can only be nonzero for positive values of the random variable. In other words, the PDF is bounded on one side. Third, the mean value and the peak value of the PDF are generally different [16].

The gamma distribution is used to model waiting times or time to complete a task. More specifically, it can be shown that if we have exponentially distributed interarrival times with mean $\mathbf{1/\lambda}$, the time needed to obtain $\mathbf{k}$ changes distributes

according to a gamma distribution with $\alpha = \mathbf{k}$ and $\beta = \mathbf{1/\lambda}$. α is called shape (or order) parameter; β replaced with (**d- a**) in this case called the scale parameter.

IV. EnhnacedEste Prototype Framework for Estimation

An advantage of modeling estimation processes through gamma processes is that the required mathematical calculations are relatively straightforward. Therefore gamma distribution with density function is recommended to use. The gamma function is an integral relationship function with gives a more accurate figure with very little variance.

A. Gamma Distribution Function

Gamma distribution with density function is recommended to estimate duration by Lootma. The expected duration should not be below the most likely duration and introduced most pessimistic duration estimate **b**, which would never (in roughly 1% of all cases) be exceeded. The formula for expected duration as given below. The values of most likely duration and pessimistic time for an activity are collected from the project manager or project team who are working on the project based on past experience [7, 19].

Activity Duration: [Most likely duration (in days) + pessimistic (in days) * 5]/6.

$$f(\mathrm{d}) = \begin{cases} \dfrac{\lambda^{\alpha}(\mathrm{d}-\mathrm{a})^{\alpha-1} e^{(-\lambda(\mathrm{d}-\mathrm{a}))}}{\Gamma(\alpha)}, & \mathrm{d} \geq \mathrm{a} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The **d** denotes the stochastic activity duration and **a** denotes the most optimistic duration estimate which would almost always (in roughly 99% of all cases) be exceeded. The shape parameter α and the scale parameter λ must be positive. If $\alpha > 1$, the duration **d** has a unique mode, which is set equal to **m**, the most likely duration estimate, so that α is called shape (or order) parameter; β replaced with (**d- a**) in this case called the scale parameter.

$$\mathrm{m} = \frac{(\alpha - 1)}{\lambda} + \mathbf{a} \quad (2)$$

Moreover, the mean activity duration is:

$$\mathrm{E}(\mathbf{d}) = \frac{\alpha}{\lambda} + \mathrm{a}. \quad (3)$$

The expected duration should not be below the most likely estimate, i.e. $\mathrm{E}(\mathbf{d}) > \mathbf{m}$. Lootma finds no reason why the expectation should be above the most likely estimate, while mostly $\mathbf{b-m} >> \mathbf{m-a}$, where **b** denotes the most pessimistic duration estimate. Hence

$$\lambda = \frac{1}{\mathrm{E}(\mathbf{d}) - \mathrm{m}} > 0, \quad (4)$$

$$\alpha = \frac{\mathrm{E}(\mathbf{d}) - \mathrm{a}}{\mathrm{E}(\mathbf{d}) - \mathrm{m}} > 1. \quad (5)$$

Lootsma estimation method is similar to three-time probabilistic model where **a** is optimistic duration estimate, **b** the pessimistic duration estimate and **m** the most likely duration estimate. Lootsma introduces the most pessimistic duration estimate **b**, which would almost never (in roughly 1% of all cases) be exceeded, in the formula for the expected duration.

$$\mathrm{E}(\mathbf{d}) = \frac{(\mathrm{b} + 5\mathrm{m})}{6} \quad (6)$$

B. EnhnacedEste Prototype for Resource Allocation

A resource is defined as any variable capable of definition that is required for the execution of an activity and may constrain the project. Resource allocation is defined as the assignment of work to an individual. When too much work is required of an individual said to overloaded, while if too little is needed it is said to be under loaded. From the pool of activity-required skills and resources, their skill has been created to select the necessary resources based on skill requirements as shown in Fig. 2. Then list the resources that will be required along with the expected level of demand. Map the resources list onto the activity. The present work models automated resource allocation to avoid overload resources.

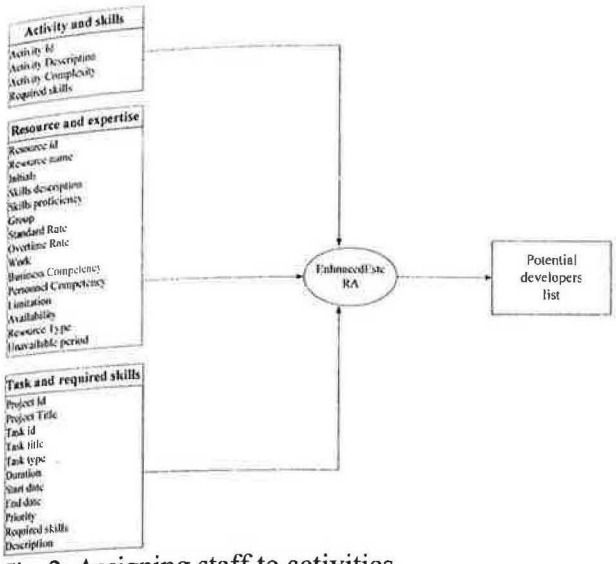

Fig. 2: Assigning staff to activities

C. *EnhnacedEste Prototype for Selecting Right Mix of Staff*

One important way to maintain sanity, and structure learning and career development effectively is to have a very clear idea of our key competencies. The review on competency regularly, with a daily review of what a particular developer has learnt and how to relates to our competencies to see what progress the developer has made. A competency is an asset of an individual who only loans it to the employer. This puts the ownership of the competency very firmly on the shoulders of the individual. There is an important distinction to be made between a competency and knowledge. Competency is something you can do and knowledge is know-how. The four important elements of Competency Measurement Model are shown in Fig. 3.

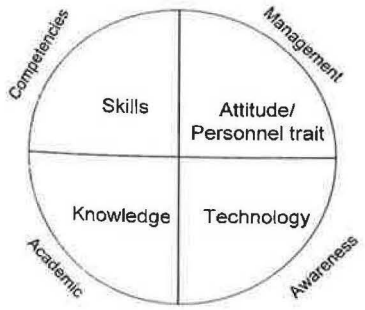

Fig 3: The link between Skills, Attitude, Technology, Knowledge.

Knowledge is linked to academic abilities. Knowledge can also be accumulated through day to day experience. Skills are tested by asking them to do it. The Figure 2 links between skills and competencies described as a competence is the efficient, effective and proper application of skills based on appropriate knowledge. A skill is also called an experience. The third element is personnel traits or attitudes. It is only by having an appropriately positive attitude that skills can be efficiently, effectively and properly applied. Technology refers to the new methodology, tools, and techniques useful for software development.

V. CONCLUSION

Data collection for the two stages is different. The sampling frame is based on the MCS IT project companies. First, a random sample was selected and physical data is collected. The exploratory research includes samples project collected from Kulala Lumpur and Melaka and Singapore. Additionally, project managers were interviewed in-depth for qualitative analysis.

The exploratory research is an empirical study and indicated that poor estimation is the most common problem in software project management. Based on this, the next stage survey is focused on human resource management of software project. Through reviewing and analyzing previous research, a conceptual framework is EnhnacedEste constructed. The model analyzes the relationship between estimation factors and performance. Since EnhnacedEste is interested in measuring software project duration, effort, it is necessary to define this concept, with regard to its sources, nature and consequences. Specifically, focused in its relationship with other attributes of the software product and process, are mainly productivity and quality.

The expectations in this field of study were that to find a rather intricate picture of software project estimation, with many different views of how it is influencing project duration and software quality. There would be a wide range of methods available for measuring software project duration, but each of them measured only some aspect of software duration. Thus, the prospect of finding one method or measure that always encompasses accurate estimation was not so bright. The choice of one method would therefore be a question of priorities, and these priorities had to be based on the prerequisites at EnhnacedEste.

VI. ACKNOWLEDGMENTS

The authors would like to thank the software house for providing raw data for this evaluation. Special thanks are contributed to Pusat Komputer, UTeM for many helpful suggestions while condensing the results. The authors acknowledge, with gratitude, the assistance and cooperation provided by dean and deputy dean of Faculty of Information Technology and Communication, during this study and for permission to publish the findings in this research paper. Million thanks go to Dr. Nanna Suryana for his guidance.

VII. REFERENCES

[1] Chase, R.B. and Aquilano, N.J., *Production and Operations Management: A Life Cycle Approach*, (Irwin Homewood, IL, 1989).

[2] Gregory, A.H., Simulating software development processes (CAPI, 1996) pp.73-78.

[3] Merran Evens, Nicholas, H., Brian P., *Statistical Distributions* (3rd, Wiley-Interscience, USA, 2000) pp. 98- 105.

[4] Pandian, C.R., *Software metrics: a guide to planning, analysis, and application* (Auerbach, USA, 2004).

[5] Pete, M.S.,*Elements of Project Management Plan, Schedule & Control* (Prentice Hall, USA, 1992).

[6] Lockyer, K. & Gordon, J., *Project management and other project network techniques* (6th ed., Prentice Hall, USA 1996).

[7] Lootsma, A Gamma Distribution as a Model of an Activity Duration, In *Methodes A Chemin Critique*, Actes du Congres International, Vienne, 1967, Dunod, Paris, 1967, pp.219-225.

[8] Walter A. Rosenkrantz, *Introduction to Probability and Statistics for Scientists and Engineers* (McGraw-Hill, USA 1997).

[9] Carl, K. C., Chikuang, C., Thinh, T. N., and Mark, C., *SPMNET: Net Methodology For Software Management*, Department of Electrical Engineering and Computer Science, University of Illinois, Chicago 1990.

[10] Eric, H. L., & Mahashwari P.,Task and team management in the distributed software project management tool: *Computer Software and Applications Conference, COMPSAC 2001 25th Annual,* (2001), pp. 401-408.

[11] Gutierrez, G.J., Kouvelis, P., Parkinson Law and its Implications for Project Management, *Management Science, vol.37, no.8,* 1991, pp. 990-1001.

[12] Jim D, Jason, B. and Gerard J. L.,A task allocation optimizer for software construction, *IEEE Software*, 2004, pp.76-82.

[13] Jørgensen, M., and Moløkken-Østvold, K., Reasons for Software Effort Estimation Error: Impact of Respondent Role, Information Collection Approach, and Data Analysis Method, *IEEE Transactions on Software Engineering*, *Vol. 30,* No. 12, 2004, pp.993-1007.

[14] MacDonell, S. G and Shepperd M.J., Using Prior-Phase Effort Records for Re-estimation During Software Projects, *IEEE* 2003.

[15] Mehwish N.,A Survey of Software Estimation Techniques and Project Planning Practices, *Procs 7th ACIS International Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06)* 2006.

[16] Schellenberg, A., Rosehart, W., and Aguado J., Cumulant-Based Probabilistic Optimal Power Flow (P-OPF) With Gaussian and Gamma Distributions, *IEEE Transactions on Power Systems*, *Vol. 20,* No. 2, 2005, pp. 773-781.

[17] Ofer M., Tzvi R., &, Dov D.,Factors affecting duration and effort estimation errors in software development projects, *Information and Software Technology vol.49,* 2006, pp.827-837.

[18] Verner, J.M., Evanco, W.M., Cerpa, N., State of the practice: An exploratory analysis of schedule estimation and software project success prediction, *Information and Software Technology Vol.49,* 2007, pp. 181–193.

[19] Lootsma, F.A., Network Planning with Stochastic Activity Durations – An Evaluation of PERT Times, *Statistica Neerlandica, vol.20*, 1966, pp.43-69.

VIII. BIBLIOGRAPHIES

He is a Deputy Dean for Research and Postgraduate Programme and lecturer at Faculty of Information Technology and Communication (FTMK), Technical University of Malaysia Melaka (UTeM). He got his PhD in year 1996 from the Department of GIS and Remote Sensing, Wageningen Research University, Holland. He published more than 60 papers in journals and conference proceeding and three books. His research interest is in the field of GIS interoperability infrastructure, Location Based Service, Spatial Modelling, Spatial and Temporal Databases and Information Security.

Sobah was born in Kulim Kedah. She graduated from Northern Territory University, with diploma in software development. She obtained bachelor degree in Computer and Mathematical Sciences from the Victoria University of Technology Melbourne, Australia. She was interactive multimedia subject lecturer at LimKokWing University of Creative Technology in Subang Jaya, head of computer department at Taylor's College in Cheras, Stamford College Malacca. She is active researcher in software project management, published her research findings.