Issues in Software Cost Estimation

Abedallah Zaid, Mohd Hasan Selamat, Abdual Azim Abd Ghani, Rodziah Atan and Tieng Wei Koh

Faculty of Computer Science and Information Technology, University Putra Malaysia,
UPM Serdang 43400 , Selangor Darul Ehsan, Malaysia

Abstract

Software Cost Estimation (SCE) is a process of predicting the efforts and cost in terms of money, schedule and staff for any software system, Software cost estimation is an old arts come with the beginning of computer industry in 1940s and it has been developed many times until formulating function points by Albrecht in 1979.

Nowadays software cost of estimation become a complicated branched science hence many functional sizing techniques, sizing metrics, cost and effort models appeared which probably not exist in this volume in the rest of sciences, this paper shows the common techniques used in SCE in general in addition it highlights the most important trends in this field also it show the urgent topics to be investigated and the challenges in SCE process.

Keywords: *software cost estimation, software cost estimation techniques, software cost estimation issues, software cost estimation challenges.*

1. Introduction

SCE is a process used in software development industry to estimate or predict the resource, efforts, cost of any development process, furthermore to the management controlling and monitoring process over the software development process, before inventing the techniques of estimation in the beginning of 1970 the software estimation process was mainly depend on a rules of thumb and some simple algorithms to estimate the size, efforts and cost after that the idea of function points was introduced by Albrecht [1] which till now gain a high reliability in the estimation process, since that time many studies carried on function points to calibrate its weights, modify its parameters and its adjustment factors to reflect the current improvement in technology and software development industry.

The history of SCE begin early in 1960 when Frank Freiman develop the concept of parametric estimation models that's leads to the development of PRICE model for hardware, in 1970 the researchers in this field analyze many projects using statistical techniques attempting to identify the factors which affect the cost of software development using correlation and regression techniques, by the end of this decade the Constructive Cost Model COCOMO is being formulated by Barry W. Boehm and C. Abts[2], PRICE software cost estimation parametric model was developed by the end of this decade also by Frank Freiman and Rbert park, Allan Albrecht and John Gaffney of IBM developed function point analysis FPA to estimate the size and effort of information systems, Halasted define another measure for software size depends on number of operands and operators which did not last in use too long.

In 1980 many evolutions occurred to the previous techniques and models, Barry Boehm and W. Royce[3] defined a revised model to adopt ADA programming language named Ada COCOMO, Capres Jones [4] improve the functionality of FPA via including the effect of complex algorithm in the computation of function point, Charles Symons [5]propose another version of FPA to handle the problem of subjectivity in FPA called Mark II Function Points.

In 1990 the leader of software cost estimation researcher Barry Boehm et.al [6] reformulates his model into COCOMO II which consist of three main submodels Called Application Composition, Early Design and Post-architecture models where he used many software sizing models like Object Points, Function Points and source line of code.
In the last decade until this year 2008 researches in SCE become numerous side to side with the rapid exponential improvement in software and information technology industry, mainly the researches related to handle the new phenomena in software engineering such as Object Oriented environment, agile projects, reusable component, SPI, Component Based Programming, real time systems etc via new sizing techniques, inventing

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

some new measures to measure the new existed items and improving and calibrating the previous models and techniques to be applicable to the new current environment, figure 1 show the main phases in SCE process over the previous decades.

Currently, SCE has gain more attention to produce a reliable trusted models which able to predict the cost and efforts to develop any software system hence the estimation process yields important results for the managers, developers and users that's allocate and determine the cost, resources and the schedule for any proposed system.

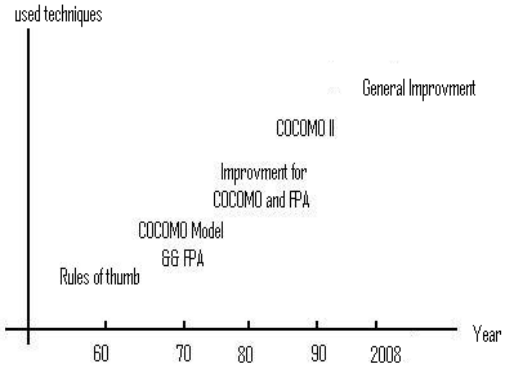

Fig. 1 main historical phases of SCE

2. Software Cost estimation Techniques

Software cost estimation can be approached and investigated in many different ways, there is mixing between the concept technique, method and even model in the research papers mentioned in the literature, so it's necessary to make a distinction for all of these definitions.

There are many classifications for the methods, techniques and models used in estimation process, Briand and Basilia and Thomas [7] classify the software cost estimation into six categories like model based, expert, learning oriented, dynamic based, regression based and composite Bayesian. Evans and Lanham and Marsh [8] propose 10 methods to estimate costs and consider each one a category according to the authors.

we recommend to make a classification of SCE where the techniques classified into qualitative and quantitative techniques, the qualitative techniques are further subdivided into intuitive and analogical techniques and mainly depend on similarity and comparison , and the quantitative ones into parametric and analytical techniques where its depend on the detailed design and can be used in advance phase of the implementation, we recommend select the techniques according to the availability of data and the environment, or it may be classified statistical or numerical techniques and the second is artificial intelligence techniques.

The current trends in utilizing the current available techniques is the ability to combine the techniques "hybrid technique" together to get an optimum predicted values in the estimation process, Aggarawl et.al [9] use linear regression techniques and neural network to predict the LOC size from the function points, Xishi et.al [10] use neuro-fuzzy approach to build Constructive Cost Mode l(COCOMO).

3. Current open research in software cost estimation

The software cost estimation process is consider a fertilize field to investigate it, since it has a main role in decision making at all levels managers, developers, users "customers", for this extra importance reasons, software estimation process should have extra efforts from the researcher and it need extra cooperation from the industrial software development company to provide the researchers by the necessary data to propose and develop a suitable proper models for the estimation process at all, here is some of the hot topics which need more focus investigation from the researchers:

3.1 Metrics conversion:

Since the nature of estimation process in software industry is differ from the rest of science branches, it yields a lot of sizing techniques "LOC, function points, object points, application points, usecase points" unfortunately there is no specific rules to convert between these metrics, Capres Jones [11] show the problem as "if the estimation process for system x using function points yield 100 function points while for same system x using usecase points yields 140 usecase points, is 140 usecase points is equal to 100 function points ?".

answering this kind of question requires a lot of industrial data for thousands of software projects to verify from the consistency of the current used sizing techniques, metrics and make sure that the estimation process is consistent using more than two different sizing techniques, this issue if it can be solved it will improve the correctness rate for the software estimation process at all and eliminate some not useful sizing techniques furthermore it increase the accuracy of the estimation process since it can be carried on in more than one metric.

3.2 ***Sizing from requirement & design phase:***

Software estimation applied at early stage of software development, so capturing the estimation from the known characteristics of software will be a useful approach for estimation, many tools used to represent the requirement and the design of any software such as UML diagrams, Z specification language, state transition diagrams, data flow diagrams.

Proposing a new functional sizing techniques based on one of the known characteristics of the requirement & design phase will produce an efficient new sizing techniques, in order to increase the robustness of this new functional sizing techniques its advised to build a sizing techniques based on two or more different well-known tool, for example propose a hybrid techniques based on the requirement and design phases such as use case and class diagram and make use of information which is available in this two diagrams, handling it in someway to compute the cost estimation of the software.

3.3 Complexity analysis of software systems:

Complexity is a variable and subjective idiom can be viewed in many different ways, simply the more complicated software yields more defects, more difficult to maintain and update, in software industry the most used formula to measure the complexity of software code is the Cyclomatic and Halasted formula.
Many sizing techniques applied in software estimation process include a lot of adjustment factors like function point, usecase points, the value of these adjustments factors assigned according to the experience of the estimator with some other defined rules, no complete specified rules or standard control the assignment process of these factors rather than these factors don't use any of the complexity measures to assign its values, so more research should concentrate to investigate the proper way to assign the adjustment factors values by highlight a standard to control this process.

Solving the subjectivity problems for the adjustment factors and the complexity factors in the sizing techniques will lead to improve the correctness rate for those sizing techniques. Al-Hajri et.al [12] and Wei, Luiz [13] produce accurate results via function point by establishing a new weights system for function points using artificial neural network and fuzzy logic.

Software engineering as a science get affected in the other branches of sciences, there is many complexity measures used in the science like Combinatorial complexity, Computational complexity, Data complexity, Diagnostic complexity and many others, more research should investigate and focus on the available complexity metrics for the other science and test which one could be added and used efficiently in the software engineering in general and in SCE specifically in a manner that can be used with function points and the other sizing techniques correctly.

Complexity issues can be improved if it can be directed to the open resource software to get deeper understanding for the pattern and complexity issues.

3.4 Risk & ROI "Return Of Investment" analysis and software project cost estimation

In software development process many projects failed or overrun its budget and resource for common known reasons occurred in all the development companies, so more research should take in consideration these common factors and try to include it in the estimation process with a margin of occurrences possibility to avoid overrun in the estimation.
Capers Jones [11] suggest ROI as a general concept used in economics to measure the profits of any products and its revenue after launching the product in the market, same in software industry the estimation process should calculate and predict the ROI as a metrics for any software before the development take place so the management and the customer can decide upon this new suggested metrics how much it will cost as a management and as a customer how much they will bid for this software.

Developing the suggested metric ROI-Point for example should measure not only revenue and cost, but it should go beyond that to measure the criteria which affect in indirect way as customer satisfactory, software quality, maintenance, operation cost, software improvement on the organization, reliability and many other where this suggested metric should let a free variable which should be assigned by expert estimator to customize this value according to the experience of estimator for both side customer and the software development company.

3.5 Investigating Open Source Software OSS

As the wheel of Open Source Software OSS technology run, Asundi [14] suggest that there is a need to develop a model which able to estimate the efforts, cost for OSS, hence the normal cost, efforts models inadequate for this purpose, such kind of models should care for the nature of OSS environment such that the development of OSS can be achieved almost 24x7 between the volunteers around the world, also how to measure the impact of the community surrounding the core development team,

rather than the normal effects of a normal software wanted to be estimate.

Investigating OSS has another face, the researches on OSS should concentrate on investigating the current available open source system by collecting the data and analyze it to identify some interesting pattern then modeling them deterministically or statistically and coming up with deep insights of OSS in terms of cost estimation process.

For example the operating system Linux, it can be investigated to check the relation between the size complexity and the efforts needed to update it as long as the operating system Linux is growing and being updated, rather the ability to check many different forms of complexity on OSS.

3.6 Revaluation of Function points:

Function points is one of the most accepted and robust sizing techniques used in the software cost estimation process, function points which formulate by albercht was established in the early of 1970 and since that time it did not modified to reflect the current statues of technology, many research held to calibrate the weights of function points [12], [13] these kind of calibrations improve the accuracy of function points and it show a need to a revaluation model for the function points.

Revaluation of function points needs lots of numerical data taken from large suitable projects from all the word to reflect the international view of the current technology, rather than the model should be customized to accept any local data, then the model automatically generates the weights for this local data.

Any evaluation of function points should specifically evaluate the 14 General System Characteristics GSC to check if it really reflects the currents statues of software development technology if you know that it doesn't take into account the environment variables for determining efforts required to develop a system, for example ISO standard function points did not include the GSC in its calculation, another suggest that some other factors should be added and another should be deleted from the 14 characteristics, and for each one a mathematical, quantitative, statistical models should be presented to prove any possible recommended changes.

Another topics to be investigated in the evaluation of functions points is test if all function points have same size, some function points for example need 30 hours per staff to be accomplished another may need 40 hours per staff, this issue should be tested to check if this variance come from the different size of function points or if its related to the variability in productivity rate correlation.

Function points proof its accuracy since decades but some research should be oriented to investigate the accuracy of function points and relate it to the programming language used in the implementation, for example .Net IDE provide many facilities in developments which increase the productivity rate of developers not like some other used languages, so there is a need for an adjustment factors which take the programming language used in development in consideration of function points estimation process, this research need a lot of historical information regards a projects developed in many languages and statistically build an identifier such as Backfiring index that compute function points based on LOC mathematically, this proposed identifier need a continuous evaluation to adopt any changes in the programming languages releases.

Function points should be expanded in away to adopt and cope with real time system and handle the algorithms for the estimation process purpose like COSMIC-FFP (Common Software Measure-ment International Consortium – Full Function Points) that can be used for sizing real time system, in addition to business software, this kind of sizing techniques represent the second generation of function points, it has some limitations and need more research to be expanded to adopt the variety combination of business and real time software.

3.7 Other issues.

Software cost estimation needs more efforts from the researcher either in academic or in industry field to cooperate together to enhance the estimation process and produce the best results, software cost estimation currently need a fully systematic review for the available research and classify it according to its content and trends, this will ease the research process and can clarify the shortcomings in this field same as what Jorgensen and Shepperd[15] did, this kind of review papers should highlight and specify what are the software cost estimation fields needed more research and study to enhance the process at all.

more research needed to sizing the software functional requirement directly once it stores in CASE tool Azzouz and Abran[16], which will leads to quick estimation and lower cost, this automatic tools enable estimators to control some of the variable values depend on the expert estimators judge.

One of the most important responsibilities lie one the industrial software development company to archive

there projects and build a data base include all the data related to each projects such as the language used in development, personnel's number in each development phase, schedule, cost, LOC, SLOC, number of classes, number of use cases, the estimate cost of the project, cost of each phase in the development, actual cost and many other factors to achieve a highly trusted historical data base and share it with the other, clearly the ISBSG(International Software Benchmarking Standards Group) benchmarking dataset show the improvement achieved in the estimation field, many models and sizing techniques can be derived and improved if these kind of historical data available, currently ISBSG in a continuous improvement, version 10 of ISBSG benchmark data released recently, another consortium including SEI(Software Engineering Institutes), SPR(Software Productivity Research) and David's consulting Croup begin a work to collect an industrial benchmark data [11] which will lead to better and highly improvement in this filed.

Handling the missing data in the historical dataset consider one of the topics that's deserve more research hence mainly software cost estimation field depend on the historical data, usually this data is not complete and missing so there is a need to suggest a certain mechanism to handle it same as what Strike and Emam did [17].

There is a need to improve the well known software cost estimation models as COCOMO, SLIM, PRICE-S models, most of these models require a historical data to forecast SCE, COCOMO model for example was published by Barry W.Boehm's in 1981 Since that time it was calibrated and modified via fuzzy logic, neural network and some other techniques to adopt the changes in the software industry and project management, for example see[10], there is a need for a framework which able to handle any changes in the calculations of these models to be up to date with the current technology.

Another urgent need in information business world today is the accurate estimates of web projects, there is a need to collect information about the web projects which developed using conceptual modeling techniques then measure it using the functional size measurement methods for web application such as OOmFPweb to estimate project indicators such as size, effort, cost and duration.

Finally, we call for increasing and concentrate about this kind of research paper hence it open and ease the way in front of the novice researchers and others to invade the software cost estimation fields rather than it enrich the research and contributions to reach a high level of robustness, certainty and accuracy in the software cost estimation, table 2 show a summary by the open researches.

Research issues
Metrics conversion
Software sizing from requirement and design
Complexity analysis
Risk and ROI analysis
Investigating OSS
Revaluation of function point
Sizing software from CASE tool
Improving SCE models
Handling missing data in dataset
Estimating web projects

Table 1 show the hot topics in SCE

4. Challenges of Software Cost Estimation.

Software cost estimation process face many difficulties to get a proper and accurate estimate for many reasons, since software is something intangible, rather than the estimation process in nature is not easy especially with intangible products, dealing with software estimation is completely different from any another estim-ation process in any fields.

One of the main difficulties in the software cost estimation process is the availability of data which is needed to verify the correctness of any suggested models, metrics and functional sizing techniques, many of the models and sizing techniques proposed based on a small amount of data, some models for example compute the size and estimates the cost based on 30 UML files, so the resulted model in this case doesn't has a high reliability and cant be generalized so it cant be applied in any development environment, even the data is very sensitive since one sizing techniques may produce a good estimate in X software development company which reside in X country but cant gain same correctness rate in Y software development company in Y country, this issue highlighted by Wieczorek and Ruhe [18], the other face of this point is the shortage of data early in the software process and this can be solved by collecting data from numerous number of projects and generalize one rule for the estimation purpose.

One of the most beneficial Benchmarking dataset is the ISBSG which provided industrial large amount of data about a completed software projects, the available data is almost completely related to tailor-made software. The "New & Enhancement Projects" Release 10 contains data of over 4,000 projects from different countries mentioned by Dekkers [19].

There is a problem in the estimation process in holding the proper and suitable data set to validate any kind of metrics, sizing techniques or any cost models, this produce a real problem in improving software estimation process, there is ISBSG benchmark data set, clearly it improve the estimation process in many fields, but still not enough, estimation area need a lot of detailed data.

another difficulties in the estimation process is the instability environment in the technology world, in a simple way "What is correct today may not tomorrow!" this clarify the rapid improvements and changes in the development process where most of the models need to be calibrated and modified from time to time to be consistent and applicable under the spot of the current development lights, and this require some variables factors in any suggested model.

Furthermore, the nature of the software development process where not all the requirement is known, in addition to requirement creep problem, also the interconnection between the cost factors and how each factor may affect the final result of estimation and may affect on some other factors, the software development field is very complicated area, as of 2007 the software industry use about 600 different programming languages, creating about 120 different kinds of different applications, software industry employee about 90 specialists, many kind of complexity, perform 23 different maintenance activities, 18 different testing activities, use 43 different design methods and use 38 different size metrics [11] These numbers show the complicated environment of software development and the interconnectivity between its elements and show that the estimation process stills a hot area for the researchers to produce more robust accurate models.
Another problem is the sensitivity of the data used in verifying any sizing techniques, metrics or models. [18] show that the effects of data used either it's company-specific local data or company-external global data showing the differences in the results which coming from the data used in the estimation process, so when the researchers formulate any estimating techniques they should care for the type of data used to formulate the model or technique.

Software cost estimation has another problem in its procedure where its need a specialist experts to make the estimation process in order to achieve accurate expected estimate, some times counting the number of function points it self may cost more than the actual development [11].

Another challenge in software cost estimation is that there is no specific rules and standard control the whole process of software development and specifically in the cost estimation process; it still an ad hoc process doesn't obeyed or restricted to a certain standard or rules, so there is a need of some approved rules and standards to control the estimation process at all.

5. Conclusion

SCE is a critical, effective process in software development and project management, many decisions stopped according to the results of the estimation, software cost estimation needs extra efforts and cooperation from the academic researchers with a help from the industrial software development companies to achieve highly trusted cost models via exchanging expertise, models of development in addition to the software engineering best practices applied in the industrial software development company and the needed suitable data to formulate the metrics and cost models in software cost estimation process.

This work highlights some of the hot topics in the estimation field, for sure there still many others not mentioned here need more researching and investigation, still there is a need to research more about the effective beneficial hot topics in the efforts and cost estimation field to open the doors for the novice researcher and concentrate efforts of researchers to improve and enhance the software cost estimation field at all.

References

[1] A. J. Albrecht, and J. E. Gaffney, "Software function, source lines of codes, and development effort prediction: a software science validation," IEEE Trans Software Eng. Vol.SE-9, pp.639-648, 1983
[2] B. Boehm, and C. Abts, "Software Development Cost Estimation Approaches – A Survey1", University of Southern California.
[3] B. Boehm and W. Royce, "Ada COCOMO and the Ada Process Model", s.l, DTIC, 1992.
[4] Capres Jones, "The SPR Feature Point Method," Software Productivity Research, Inc, 1986.
[5] Symons, Charles, "Software Sizing Estimating: Mark II FPA," Wiley, 1991.
[6]. B. Boehm, B. Clark, E. Horowitz and C.Westland,"Cost models for future software life cycle processes: COCOMO 2.0", Springer Netherlands, Annals of Software Engineering, vol 1, pp. 57-94. 1995.

[7] C. Briand, R. Basili, M. Thomas, "pattern recognition approach for software engineering data analysis," IEEE Transactions on, vol. 18, pp. 931-942, 1992
[8] D.K. Evans, Dr J.D. Lanham1, Dr R. Marsh," Cost estimation method selection: matching user requirements and knowledge availability to methods", Systems Engineering and Estimation for Decision Support (SEEDS) Group, University of West of England, Bristol, UK.
[9]K. K. Aggarwal, Y. Singh, P. Chandra, M. Puri. "An Expert Committee Model to Estimate Lines of Code"ACM ,vol 30,pp 1-4 ,2005
[10] X. Huanga, D. Hob, J. Rena and Luiz F. Capretza," Improving the COCOMO model using a neuro-fuzzy approach," Journal of Systems and Software, vol.7, pp. Pages 29-40, 2005.
[11] Capres Jones, Estimating Software Cost, New York : McGraw-Hill.
[12] M.A. Al-Hajri, A.A.A. Ghani, M.S. Sulaiman, M.H. Selamat," Modification of standard function point complexity weights system," Journal of Systems and Software vol.74 ,195–206, (2005)
[13] Wei Xiaa, Luiz Fernando Capretz," A new calibration for Function Point complexity weights,": Journal of Systems and Software, , vol. 50, pp.670-683 ,2008
[14] J. Asundi. " The Need for Effort Estimation Models for Open Source,",ACMvol30,pp1-3,2005.
[15] M. Jorgensen,M. Shepperd,"A Syestematic Review of Software Development Cost Estimation Studies,", Software Engineering, IEEE Transactions on, vol. 33. pp 33-53, 2007.
[16] S. Azzouz, A. Abran, " A proposed measurement role in the Rational Unified Process (RUP) and its implementation with ISO 19761: COSMIC FFP," SMEF., Italy , 2004.
[17] K. Strike, K. Emam," Software Cost Estimation with Incomplete Data,", IEEE, vol. 27, pp. 890-908. 2001
[18] I Wieczorek, M Ruhe," How valuable is company-specific data compared
to multi-company data for software cost estimation? "METRICS.02 , IEEE, pp 237- 246, 2002
[19]. T. Dekkers ,"Benchmarking is an essintial control mechanism for managament,", RPM-AEMES, vol. 4,pp 99-103, 2007.

Abedallah Zaid received his bachelor in software engineering from HU Jordan in 2005 and his master degree is in software engineering from University Putra Malaysia in 2008, currently he is pursuing his PhD in software cost estimation and software sizing measure.

Mohd Hasan bin Selamat received his M. Sc in Computer Science from Essex University in 1981 and M. Phil in Information System form East Anglia University in 1989. His research interest includes component-based software development, knowledge management, software costing, and strategic information system planning. He is now an Associate Professor in The Department of Information System, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. He has published a number of papers related to this area.

Abdul Azim Abdul Ghani is an Associate Professor cum the Dean of the Faculty of Computer Science and Information Technology, University Putra Malaysia. He obtained his PhD in computer science from University of Strathclyde, Scotland. His research interests are software metrics and software quality.

Rodziah Atan received her PhD from University Putra Malaysia in 2005. Her field of interest is software process modeling and pursuing for new knowledge in bioinformatics visualization tools and other software engineering related areas especially software design. She is now appointed lecturer and researcher in University Putra Malaysia since 2006.

Koh Tieng Wei received the first degree in computer science in 2003 and the MS degree in software engineering in 2006 from the Putra University, Malaysia. Currently, he is pursuing a PhD degree with his research work related to object oriented software sizing measure.