A Man-hour Estimation Tool Focused on Graphical User Interface

Yoshiharu Imai, Yoshikatsu Ohta, and Hidetomo Suzuki

***Abstract*—Purpose of this study is to develop a software development man-hour estimation tool which gives same result without regard for estimators' experience.**

This study is based on an idea that there is no "one true value" of software development man-hour estimation or it is too difficult to find the value because every software development teams are different on experience and technique. This study focused on finding an estimation value which every engineers consent to use or at least to stop protesting against to use.

In order to find the value, a questionnaire of software development man-minutes on each component of GUI was implemented using both of WEB site and papers.

This study's target software development estimation tool was made as a spreadsheet with pairs of component name and value and everyone get same estimation result by using it only counting number of components on screen design documents and filling numbers of components on the spreadsheet.

***Index Terms*—Software development cost, estimate, graphical user interface.**

I. Introduction

Generally speaking, persons who have authority of decision to do a software development project or not, such as end users or manager of business, care necessary length of period, number of staffs and budget of software development project and quality of software product which is generated by the project [1].

Most of estimate problems which happen at software development project such as estimate error and different results of estimate are avoided if both of software developers and clients can estimate those things and reach same results.

It is correlative in number of moving data and software scale, and screen design documents are provided in many cases at an early stage of developing interactive software with highly sophisticated Graphical User Interface (GUI) [2]. Therefore, we can estimate software development cost by estimate GUI development cost at enough early stage of software development project.

Recently, techniques to develop software systems have been become complicate more and more mainly on a part about the security. Especially WEB application systems which everyone try to access on public network became majority of software development projects and latest software development project became had to include more sophisticated techniques than traditional software development projects which are used only in a closed network. However, it does not change from traditional way, except to develop security techniques itself, that main factors of software development cost are system requirements and specifications because developers can easily include available component of latest security techniques into software systems by support of sophisticated integrated development environments.

This study aimed to solve estimate error problems by developing a software development man-hour estimation tool which focuses on GUI and gives same result without regard for experience of estimators which include end users who do not have a technical background.

Manuscript received October 5, 2015; revised February 23, 2016.

Yoshiharu Imai, Yoshikatsu Ohta, and Hidetomo Suzuki are with Division of Information Engineering, Graduate School of Engineering, Mie University, Tsu, Japan (e-mail: imai@net.info.mie-u.ac.jp, ohta@net.info.mie-u.ac.jp, suzuki@net.info.mie-u.ac.jp).

II. Background

A. Meaning of an Accurate Estimation

It is necessary to know exactly beforehand how long and how many people are necessary for developing target software system in order to succeed software development project. In order to know it, many kinds of estimation methods are devised. However, any estimation method which can reach to same value by everyone and are enough effective were not devised yet.

Actually, it is difficult to make a software development project succeed without checking progress of the project and taking measures to delays of it even though a project manager of the project can get a right estimation of software development cost at the point of start. However, no project manager can manage any software development project effectively except accurate estimation and accurate estimation is one of necessary condition to let project managers manage their software development projects.

B. The Present Conditions of the Software Development Estimate

On software development project, average excess rate of schedule was 120% and average excess rate of budget was 100% [3]. Moreover, it is expected that there is more excess judging from a planned plan at first by these results because these results were calculated after those projects' scales were reduced in order to make them within schedules, budget or both [3].

In other words, in average, software development projects could not satisfy the planned function at first even though those projects exceeded 2.2 times in schedule and twice in budget from the estimate. We cannot call these projects meaningful plan anymore.

Causes that these exceeding problems are frequent are that most projects were planed and carried out under excessive pressure on schedule and most engineer could not

doi: 10.18178/lnse.2016.4.3.245

persuade stakeholders by providing explanation based on scientific grounds or fact at least enough powerful to save plans of projects.

C. Validity of using GUI for Estimate from Estimate's Scope Point of View

Fig. 1 is Boehm's chart titled "Software cost estimation accuracy versus phase" [4]. Validity of estimate depends on software definition details level and it can be confirmed in this chart. Boehm's chart shows relationship between software development projects' mile stones and scopes such as estimate of functions, man-hour and cost. However, the chart shows relations when a project was managed in an ideal state to the last. In other words, no one can estimate enough accurately to control a software development project during lower details level, even though they could define ideally the target software system.

Furthermore, according to this chart, a minimum and a maximum difference are 16 times in the initial concept. Steve McConnell said "a skilled project manager can navigate a project to completion if the estimate is within about 20% of the project reality.", and also "Meaningful commitments in the early-middle part of the project (about 30% of the way in) are possible and appropriate." [3]. In other word, this means that making commitments at the stage of completing user interface definition is appropriate, and this idea suggests validity of using screen design documents to estimate software development cost in this study.

III. Related Studies

Estimating cost of software development project become difficult by traditional ways of software estimate, such as number of steps because after 4th generation language, development tools' evolution have made impact factors of necessary cost of software development be not only making algorithm but also tweaking properties as programmers' skill.

On the other hand, other estimate methods such as function point are studied, but those methods require estimators' skills such as predicting internal logic and experiences of software development. Therefore, these estimate methods are not enough to persuade end users of accuracy of the estimate.

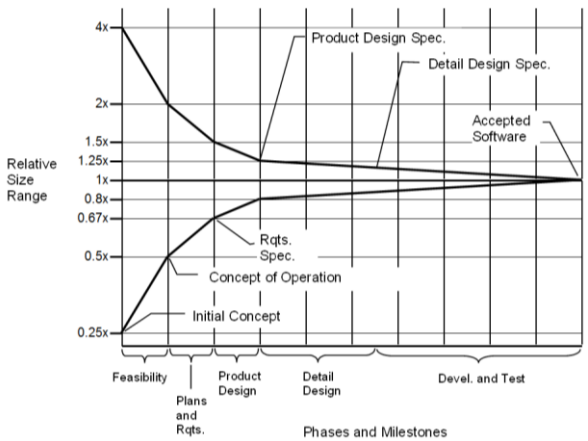

Fig. 1. Boehm's "Software cost estimation accuracy versus phase".

Recently, an estimate software cost from screen design documents with The Common Software Measurement International Consortium - Full Function Point (COSMIC-FFP) method was studied [5]. However, it was not enough to estimate automatically by end users because unevenness could occur in the estimate by estimators' decision of reusing components and unsuitable result could be given from a real reuse degree.

Meanwhile, it has been said that collection of past data was the most important to accurate estimate in studies of estimate [1]. There is difference among project on software development ability of team members, and it is self-evident that using past data is the best way to estimate software development cost by same team.

However, for example in Japan, the spot in the front line of the software development is not supported by engineers of original contractor, but is supported by engineers of the subcontract companies. Actually, it often happens that only the nominal leader of project from the original contractor, an engineer of sub-subcontractor or sub-sub-subcontractor becomes the substantial leader of development team and all other members also are from non-original contractor companies. Therefore it is common in Japanese development spot that development members are different every project. It is impossible under such situation to predict the ability of the development member from the past results.

As a result, it is necessary to make the guidance measuring the average and common development talent in particular area which is not for a specific development team as a general-purpose estimate tool. Even though we researched to Japanese engineers, estimate method in this study is not only good for any areas where development team members are not stable, but also good for any areas where any software development estimate tools for non-experts are not exist.

IV. Software Cost Estimate in This Study

By using an estimate method in this study, authors succeeded to persuade end users agree with the author's estimated schedule in the past project when end users required too short schedule to develop a software system with highly sophisticated and complex GUI, because it could be used by even end users without any technical background. However, the estimate method could not be said a general-purpose tool at that time because the standard of the estimate every GUI components based on skills of the author and other project members.

V. Method of the Study

A. Purpose of the Study

The purpose of this study is following two points.

- To develop estimate method which non-computer experts also can estimate software development cost in man-hour from screen design documents.
- To digitize experts' estimate used for validity inspection of various estimate methods and to provide it to practical use.

B. Target Beneficiaries of the Study

This Study Aimed to Satisfy the Needs of 2 Next People

- Software developers who estimate software development cost as an important step to realize users' requirements as a software system.
- Stakeholders who want to know the function scale of software delivered as an important factor to measure result of the development.

C. Target Software System of the Study

Target software system of this study was not systems which need the high security and accuracy such as nuclear power plant, but commercial systems which have GUI for the unspecified number of users.

D. Procedure of the Study

This study aimed to generalize the method by calculating software development cost using following procedure.

- To make a list of all components that constitutes GUI.
- To calculate developing cost in man-minutes on every single components.
- To calculate software development cost of whole module by crossing number and cost on each components and total up them.

We categorized GUI's constituent elements to 43 groups as units which realize target function. After that, we re-categorized those 43 groups into 11 categories such as following list.

- Input and output components of letters and graphics
- Tables and selecting components
- Other components
- Processes not to depend on a component
- Setting for a whole window
- File manipulating
- Fonts/colors settings
- Printing
- Print preview
- Print settings
- Error Indication

In this study, we used the estimate by experts which used in inspection in precedent studies for calculation of necessary development cost on each GUI components. We made questionnaire survey for engineers who match this study's condition, and calculated the mean from the result except extreme deviations from the mean.

VI. Result

We picked up 689 company names as software developers in company data list website for Japan which showed company information along the company hope, checked all companies' own website of those 689 companies and picked up 272 companies which seems to do or to take part software development by themselves. After that, we made phone calls all to those 272 companies to co-operate this study and 82 companies agreed to co-operate. We also asked to co-operate this study to freelancer engineers of author's personal network. As a result, 217 answers were collected and there were 215 valid responses in total.

We excluded 19 data from remained data for the following reasons such as "all components were estimated same value which was 1", "all components were estimated same value which was 0" and "all components were estimated same value which was excessively large", and finally there were 196 valid data remained.

The age range was 21 years old to 56 years old and average age was 36.5 years old. The number of the information processing duties engagement moon was from six months to 500 months and was an average of 156.3 months. The number of times of upper process experience of software development was from 0 times to 200 times and was an average of 14.5 times. The number of times of cost estimating software development using integrated development environment was from 0 times to 240 times and was an average of 19.8 times. The number of the development experience moon using integrated development environment was from 0 months to 270 months and was an average of 81.4 months. The number of computery career month was from 14 months to 480 months and was an average of 217.5 months. There were 100 engineers out of the respondent with a blank after worked on software development and the number of blank month was from one month to 150 months and was an average of 35.6 months.

There are results of developing cost in man-minutes on every single component in Table I. Estimator can use the result by crossing number and cost on each components and total up them to calculate software development cost of whole module.

TABLE I: Developing Cost of Each Components of GUI

No.	Function of component	Developing cost in man-minutes
Input and output components of letters and graphics		
1	Fixed letters indicator	35.21
2	Fixed letters indicator with a hyper-link	31.85
3	Letters and numbers inputter without character decoration	42.34
4	Letters and numbers inputter with character decoration	53.04
5	Graphic tool which can drawing and indicating	59.99
Tables and selecting components		
6	Fixed table	89.00
7	Table inputter	147.10
8	On and off selector	34.39
9	Non-list type mono selector	37.51
10	Fixed list indicator	52.58
11	List type mono selector	55.03
12	List type multiple selector	70.97
13	List type mono selector with free inputter	80.41
Other components		
14	Three indicator	100.71
15	Tab type selector (sub window panel selector such as option settings)	85.88
16	Date and time inputter	49.97
17	Calendar	59.04
18	Horizontal scroll bar	39.91
19	Vertical scroll bar	40.94
20	Page change button of listed words indicator	55.03
21	Volume up/down button	45.34
22	Variable volume control bar/place controller bar of video/music player	65.13
23	Process progress indicator bar	77.15
24	Interval timer	61.95
Processes not to depend on a component		
25	Opening/Closing process of a window	71.42
26	Update/select data from a database include connecting and parse an SQL	121.60
Setting for a whole window		
27	Setting pop-up text/hints with mouse pointed on (for adding one text)	37.13

No.	Function of component	Developing cost in man-minutes
28	Sub-menu setting with cricking right button of mouse	45.07
29	Setting a menu bar on a window	41.85
30	Setting a function on a tool bar on a window	47.31
31	Setting status bar on a bottom of a window	44.69
32	Setting an icon on status bar	45.39
	File manipulating	
33	File selecting dialog	48.24
34	File saving dialog	50.68
35	Folder selecting dialog	48.40
	Fonts/colors settings	
36	Font setting dialog	40.49
37	Color setting dialog	41.40
	Printing	
38	Print dialog	44.28
39	Execute print not include making a document	64.30
	Print preview	
40	Print preview dialog	49.16
41	Indicate Print preview	67.32
	Print settings	
42	Page setting dialog	45.18
	Error Indication	
43	Error information dialog	61.32

VII. Conclusion

We developed a software development man-hour estimation tool which gives same result without regard for estimators' experience in this study. Engineers could solve problems in the spot of the software development by using this tool because it can reduce a gap of the recognition about schedule and man-hour of software development between developers and clients. For further study, we plan to collect more data to increase the precision of this tool.

Acknowledgment

We want to thank companies which agreed to distribute questionnaires to companies' engineers. We want to thank all engineers who answered questionnaires of this study.

References

[1] H. L. Putnam and W. Myers, *Five Core Metrics: The Intelligence Behind Successful Software Management*, Dorset House Publishing, NY, 2003, ch. 1.

[2] M. Nonaka, A. Kakurai, E. Bukhari, and M. Azuma, "A method for measuring functional size and complexity of interactive software based on interface specifications," *Journal of IPSJ SIG*, pp. 171-178, March 2002.

[3] S. McConnell, *Software Estimation: Demystifying the Black Art, Microsoft Corporation*, Redmond, WA, 2006, part 1, ch. 3.

[4] B. W. Boehm, S*oftware Engineering Economics*, TRW Defense Systems Group, Redondo Beach, CA 90278, 1983, p. 12.

[5] COSMIC Core Team Authors, *COSMIC-FFP Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761:2003) Version 2.2*, January 2003.

Yoshiharu Imai was born in Tokyo, Japan in 1966 and is a doctoral student of Division of Information Engineering, Graduate School of Engineering, Mie University Japan. He obtained a master's degree of engineering at Mie University, Tsu-city, Mie, Japan in 2007.

He also is the director of Applied Information Technology Laboratory. He has a wide field of activity such as technology consulting, technical writing, professional engineer training and translation. He wrote jointly a book of the title called "Sick Building Syndrome in Public Buildings and Workplaces", London, NY: Springer (DOI 10.1007/978-3-642-17919-8) in 2011. His research interest is prevention of IT engineers' death from overwork. He attempts to approach his goal by developing tools for project management such as cost estimation and quality admeasurement tools of software.

Mr. Imai is a member of Information Processing Society of Japan (IPSJ), IPSJ SIG Software Engineering (SIGSE) and also Society of Indoor Environment Japan. He won the 69th Best Paper Award of IPSJ National Convention in the article called "A Method to Estimate Software Cost Focused on Graphical User Interface" in 2007.

Yoshikatsu Ohta was born in Nagoya, Japan in 1953. He received his M.E. and Ph.D. degrees in information engineering from Nagoya University in 1978 and 1988, respectively.

Since 2004, he is a professor of the Division of Information Processing of Mie University in Tsu, Japan. During 1978-1980, he was a research associate at Mie University. During 1980-1989, he was a research associate at Nagoya University. During 1989-2004, he was an Associate professor at Mie University. His current research interests are computer networks, programming language, operating systems and term rewriting systems.

He is a member of Information Processing Society of Japan (IPSJ) and the Institute of Electronics, Information and Communication Engineers (IEICE).

Hidetomo Suzuki was born in Hokkaido, Japan, on August 28, 1957. He received his M.E. degree in information engineering from Toyohashi University of Technology, Toyohashi Japan, in 1982, and his Ph.D. degree in information engineering from Nagoya University, Nagoya Japan, in 1988.

During 1982 to 1984, he was an education associate at Toyohashi University of Technology. During 1984 to 1989, he was a research associate at Nagoya University. During 1989 to 1991, he was a Lecturer at Nagoya University. Since 1991, he is an associate professor of Division of Information Engineering at Mie University, Tsu Japan. His research interest includes image processing techniques and their application, such as medical image processing, fingerspelling recognition, and includes disaster simulation and environment recognition based on robot vision.

Dr. Suzuki is a member of Institute of Electronics, Information and Communication Engineers (IEICE), Information Processing Society of Japan (IPSJ), Japanese Society of Medical Imaging Technology (JAMIT), Robot Society of Japan (RSJ), and Japanese Society for Magnetic Resonance in Medicine (JSMRM).