American Journal of Software Engineering, 2013, Vol. 1, No. 1, 5-11
Available online at http://pubs.sciepub.com/ajse/1/1/2
© Science and Education Publishing
DOI:10.12691/ajse-1-1-2

Effort Estimation of Software Development: Depth View on IRAN Case

Yaeghoob Yavari[1], Hassan Bashiri[2,*]

[1]Department of Computer Engineering, Payam Noor University, Hamedan, Iran
[2]Department of Computer Engineering and Information Technology, Hamedan University of Technology, Hamedan, Iran
*Corresponding author: bashiri@hut.ac.ir

Received April 06, 2013; Revised November 10, 2013; Accepted November 11, 2013

Abstract Although the age of software engineering goes more than a half of a century, successful software projects which meet time and costs based on primary estimation are also in high risk. There are various reasons with different weights for that. Management problem, scheduling and effort estimation are the most important ones. Precise estimation for software development is a very hard and complex task. Because of software special problems in Iran estimation challenges have been doubled between software development companies. Based on our experience at work, many incorrect estimations in software companies urged us research about more accurate statistics and reasons for this problem. More than thirty known companies were selected. We prepared a questionnaire to collect data about companies, as well as, their projects and information about project estimation. The results were analyzed and showed that in most of the cases estimation deviation is more than 40% from primary estimation and in some cases it is up to 60%. These statistics show the necessity of traumatology and more investigation into reasons of these deviations. Our studies have specified the most important reasons of these deviations and suggest some approaches to control them.

Keywords: *software metrics, effort estimation, use case points, analogy, agile*

Cite This Article: Yaeghoob Yavari, and Hassan Bashiri, "Effort Estimation of Software Development: Depth View on IRAN Case." *American Journal of Software Engineering* 1, no. 1 (2013): 5-11. doi: 10.12691/ajse-1-1-2.

1. Introduction

In recent decades software applications have brought a revolution in the world. However software industry is contributing in almost every field of the world, at the same time it is also facing some problems due to the high failure rate [1].

In every project, handling the time and other resources are key factors to success. Different factors such as the time must be spent, the financial expense included, the resources must be allocated in whole process are estimated to conduct a project. Evidently, in software industry, because of unique characteristics of software products, addressing the issue of estimation has double significance. This significance is not a claim proposed in this text, but in referring articles, compilations and research documents of software industry this matter is clearly observed. Also, regarding the cases referred in this text, the importance of this issue is distinguished.

Metrics used to evaluate the software processes, products and services are termed as software metrics. Software metrics are techniques/formulas to measure some specific properties or characteristics of software. In software engineering the term "software metrics" is directly related to the measurement. Measurement is introduced by Information Technology Organization to better understanding, evaluate, control and predict software processes. Software measurement is essential to achieving the basic management objectives of prediction, progress, and the process improvement. Measurement as the process by which numbers or symbols are assigned to attributes the entities in the real world in such a way to describe them according to clearly defined rules [2].

Quantitative management is actually the way towards the successful project management. It applies measurement analysis to analyze product, predict the minimum possible cost and time required for software application development .Measurement analysis can be applied to all artifacts of the software development life cycle to measure the performance and improve the quality. Measurement analysis allows stating quantitative organizational improvement objectives and decomposes them down to measurable software process and product characteristics [3].

Effort estimation of software development is one the most important activities of software management and also is of the most significant reason of software project failures ([4,5,6,7,8]). Thus, effort estimation of different parts of software development affects anything from project proposal investigation, project management, analysis and design, to product quality and efficiency, customer satisfaction and success in the market.

Effort estimation is applied as an input for project planning, iterations planning, budgeting, capital analysis, pricing process and tendering. Accuracy of the software development effort estimation is one of the challenges for every software project, since it has a severe impact on expense, timing, functionality, and the development software quality [9].

While effort underestimation causes the project to face problems such as delayed delivery, budget overflow and low quality of software, effort overestimation also causes to lose market opportunities and to use resources inefficiently. The estimations influence expenses, timing prediction, performance and quality ([10,11,12]).

Planning, monitoring, and controlling software development projects need the effort and expenses estimated properly [13]. If we can exactly estimate effort for the project, project quality and efficiency are controllable, since there is no need to many changes sporadically causing the quality and efficiency sacrificed [11]. A general principle in project management is that projects must not fail in implementation phase; they fail in the planning phase. Estimation in software product development and effort is a tool to help decision making process and to provide the information needed to suggest price and negotiate with customers. Estimation helps a good commercial decision, specifies the project feasibility in a given time limit as well as functional and budget limitations [17]. In fact, software estimation in the preliminary phase of development life cycle process surely decreases the risks [13]. Rapid and accurate estimation of software projects development effort in information technology industry is determinant and fundamental [14].

Among the major problems of software industry[1] is unavailability of quantitative data. This matter hinders accurate and numerical investigation and evaluation in different parts of software development and issues are proposed generally and qualitatively. Regarding the future vision of software industry and software development pathfinders, evaluating current problems of software development effort should be dedicated mainly to accurate planning and correct analyzing the project, and with the aid of CASE tools in implementation phase, the work should be done with more speed and higher accuracy.

In the last decade, the number of software companies has been constantly growing in Iran, as SANARAY[2] institution announced that by July 2008 more than 1400 software companies were working in Iran. Although there are many companies in Iran, their staffs are not so much, as even in the largest companies, the number of the staffs does not exceed 700 people. Also various products are produced in these companies from small-scale programs to large-scale organizational programs. Some companies, as well, work on ERP products development for major factories such as automobile factory and iron-foundry [15].

2. Literature Review

Regarding estimation, numerous studies have been done especially about estimation methods. From 1979 many methods have been proposed and some references [16] have referred to more than 158 estimation methods. In this section, we consider some cases explaining estimation importance and we will provide statistics of failures arising from estimation, and then we will refer to some studies relating to Iran's software companies.

Different and many statistics of software project failures and stating inaccurate estimation as the main factor of their failure indicate high importance of accurate estimation of software [4,5,7,8,11,12,17,27,28]. In 2003, NASA repeated CLCS system [16] after spending hundreds of million dollars on software development. Initial estimation of this project was 206 million dollars, but it increased to an amount between 488 and 533 million dollars. As the result of canceling this project, about 400 developers lost their jobs [16][3].

Also, a study conducted in England shows that among thousands of delivered software products, 66% of them have overrun their timing, 55% of them have had budget overflow and 58% have unexpected major problems. Moreover, other studies show that between 30% and 70% of estimations are not correct. In 1995, Standish Group's Chaos Report showed that 89% of projects overran the resources. Other investigations show 30% to 40% of overflow in programs [4]. A possible answer to the question of why estimation matters is that estimation can help make a good commercial decision [17].

2.1. Investigating Activities of Iranian Software Companies

In 2008, a study was done on Iran's software companies focusing on investigating agile methodologies [15]. This study showed that most of the companies worked independently in Iran. Most of them were private, and few of them were public companies. Because of financial and organizational restrictions, their activity is based on internal market. Most of the Iranian software companies have small teams and one of the main reasons of this smallness is that companies often develop products with small scale which do not need large teams, and also managing the small teams is simpler [15].

In this study, according to Figure 1 which shows product scale based on subsystem, project sizes are often small or medium-size and most of the projects include less than 10 subsystems.

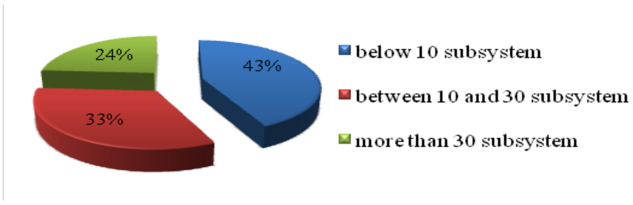

Figure 1. Product scale

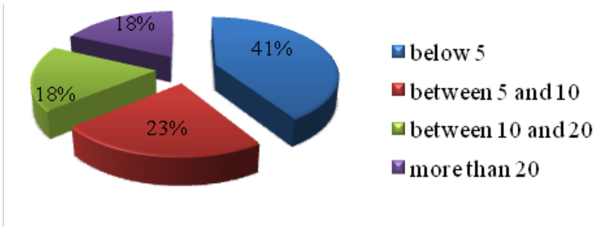

Figure 2. Team Scale

[1] Complexity in problem domain, The Difficulty of Managing the Development Process (Development problems) and using problems

[2] A consortium of IRANIAN big software companies founded at 1998

[3] See *http://csse.usc.edu/events/2007/CIIForum/pages/welcome.html*

In Figure 2, the size of teams has been shown; most of the teams working on software projects in Iranian companies have a small size with less than 5 people in each. The main reason of this has been stated as simpler management of small teams and more flexibility in face of environmental changes. This claim is in conformity with product sizes, and products smallness is proportional to team sizes.

In Figure 3, the development process used in developing software products have been shown. According to the results stated, RUP is the most commonly used one. It is noticeable that most of the companies believe that using a development methodology is necessary. Also it is remarkable that most of them use their own process model to develop, and usually XP and RUP are considered as the basic development process which they use to obtain their intended efficiency model.

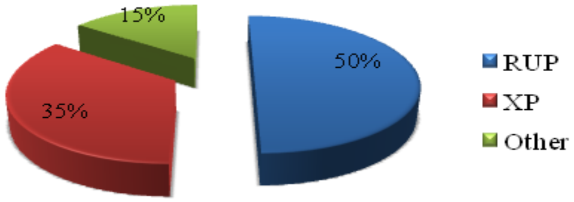

Figure 3. Development Process

The results of using development processes in Iranian companies have been shown in Figure 3; and the percentage of using agile methods as compared with non-agile methods has been shown in Figure 4. (Of course it should be mentioned that this statistics do not show the real number of agile methods use among Iranian companies; since in the study done we encountered similar claims, but after investigating companies' statements it was made clear that they do not have a clear understanding of agile methods, but that they propose such a claim mainly because of market time restrictions or to escape from documenting.)

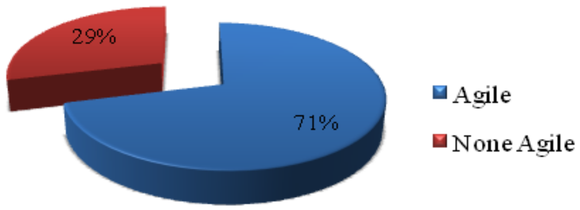

Figure 4. Agility versus Non Agility

Also this study has addressed the amount of using modeling techniques which can be useful in terms of content analysis and investigating time estimation in our research. The amount of using these techniques has been shown in Figure 5.

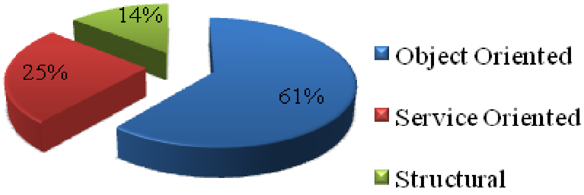

Figure 5. Modeling Techniques

2.2. Structure of Iranian Software Industry

One of the other studies on software companies in Iran has been done by Manchester University in 2003. Focus of this study is to investigate strength and weakness points of Iranian software industry export [18]. In this study, factors such as internal and religious instabilities, American boycotts, old technologies, lack of international quality standards and lack of copy write law with sanction etc. are among hindrances of software industry in Iran.

From 1960 to 1979, Iran's computer industry has had a remarkable progress and creative activities have flourished in it. For example, the first computer system with Persian language support has been developed in these very years. After the revolution in 1979, conditions changed considerably and after a ten-year stagnancy, by the end of 1990 communication and information technology realm found better conditions by reformatory changes. This study summarizes key challenges in Iranian software industry in the form of following questions:

- How can Iran sufficiently meet the internal demand for software programs both in public and private sections?
- How can Iran's software ability be developed to reach worldwide competitions?

3. Methodology of This Study

In two previous sections the reason of addressing software industry structure in Iran and Iranian companies' activities has been acquaintance with type of activity and their structure which is essential to select our research input companies. Based on this view and that there are many software companies in Iran according to the statistics published and that it is not possible to investigate all of them, we selected a number of companies in some large cities and conducted our research on 32 software projects developed by these companies. These projects with different sizes were developed in different companies having different working background and field. General information related to the projects is shown in Table 1. We collected information of thirty two projects. Projects which we gathered data about were all of regular kind like web-based, desktop application, hardware-level applications, and embedded projects and so on. To collect information, questionnaires and face to face interviews were used. In some cases, because of physical inaccessibility, the questionnaires were electronically mailed to the companies and responses were received.

Table 1. General information related to projects

Project Numbers	Average Cost (person-hours)	Average Product Scale (SLOC)	Project Type	Average Corporation History	Average Product Developed
32	17.62	124314.81	Web-Based, Desktop, System, Network, Embedded,	3.05	11.58

The questionnaire was designed in four parts. The first part is general information about the company, such as the company's background and number of the products developed. The next part of the questionnaire includes general information of the projects developed by companies; information such as project name, project type etc. The third part includes expense information of each project; information such as duration of project, number of people involved in project, project development process, product size according to SLOC, estimation method applied in project, estimators of project, the amount of agreement between real data and the data estimated. And the last part of the questionnaire includes 32 major metrics in each project. We classified these parameters into four groups including (Table 2):

Table 2. Project Features

Groups	Features
product features	Required Reliability
	Documentation
	Stable requirements
	Efficiency
	security features
	Changeability
	Operational ease
	Installation ease
	Distributed systems
	Database Size
	Language
	Reuse
	Complexity
	Software Architecture
framework features	Resource Constraints
	Platform Volatility
personnel features	Personnel Capability
	Personnel Continuity
	Personnel Experience
	Familiar with RUP
	Application experience
	Object oriented experience
	Analyst capability
	Motivation
project features	Tools and Techniques
	Risk
	Schedule Constraints
	Process Maturity
	Team Cohesion
	Multisite Development

Among these features is reliability, stable requirements, efficiency, security characteristics, reusability, object-oriented familiarity, senior analyst ability, the tools and techniques used, risks, team integrity etc.

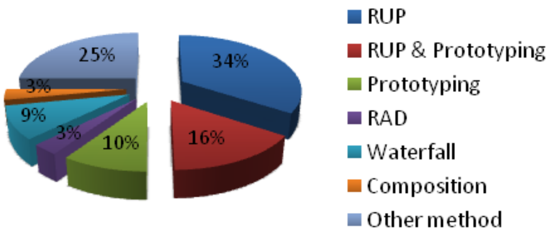

Figure 6. Development Processes

Numerous software development processes are applied in Iranian software companies and most of the companies use methodologies proportional to their situation and structure which are usually composition of standard methodologies. We have restricted development processes to some more often used ones. These processes include RUP, RUP and prototyping, waterfall method, prototyping, Rapid Application Development (RAD), mixed method and the methods other than these ones. Results from the amount of development process applications in Iranian software companies brought in Figure 6 shows that the most use relates to RUP methodology.

So far many methods of software development effort estimation have been proposed. In some texts and references (Menzies 2007) even 158 methods of estimation have been mentioned, but among these methods a limited number of them are applied as general methods in projects, and also some estimation methods have been provided for the products with a specified and special application or for use in a special company. We have specified the most common used methods in software projects (based on the statistics provided in articles and reports of project we acquired from the questionnaires) as the intended estimation methods. These methods included Function Points, COCOMO methods, Use Case Points, expert opinion and analogical method. Results of this section showed three methods of expert estimation, analogical estimation, use case points estimation or a composition of these methods were most commonly used, which was of high importance and indicated significant points to which we will refer in the next section. In Figure 7 estimation methods and usage percentage of each have been shown. According to this, the most percentage of using project sources estimation methods was related to analogical method with 29%, and the least percentage was jointly related to use case points and other methods with 12.9%.

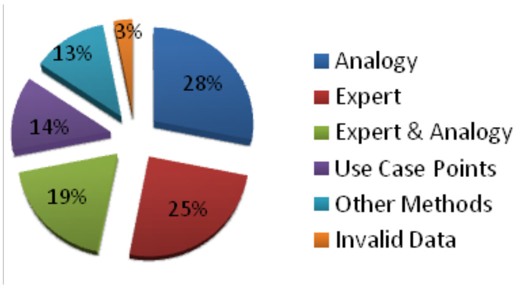

Figure 7. Estimation Methods

One of the other important sections of this study was the amount of agreement between real data and estimated data which properly shows the reason of significance of topic of this study and the reasons of necessity to consider the estimation issue. The other remarkable point is that because of deviation in software project estimations, developers pay little attention to the issue that this deviation is direct, distinct and influencing factor of expenses and time increase, product quality and efficiency decrease and as a result making loss or profitability reduction in commercial companies, and those who are aware of this matter apparently do nothing to improve it!

The information obtained from case studies shows that only in one case the estimated effort and real amount were in conformity more than 80%, but in other projects deviation was more than 20%. Also in less than 50% of projects deviation was less than 20%, and in 37% of projects the deviation was 40% to 60% which is a significant and remarkable statistics, and probability of project failure is very high in these cases. Figure 8 shows the agreement between real amount and the estimated one, and indicates the reality that the most amount of agreement between real data and estimated data has nearly been about 60% to 80%.

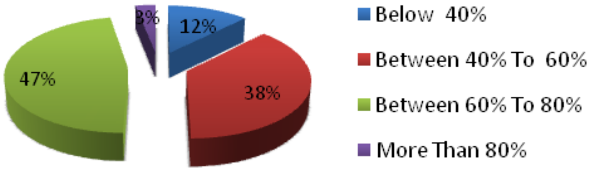

Figure 8. Amount of Adaptation Estimate Data

As it was proposed at the beginning of this article, the main issue has been to investigate reasons of problems and challenges in order to have estimation close to reality in software projects of Iranian companies. In studies, numerous reasons were recognized; reasons such as the effect of political affairs in national and international level, internal policies in industry realms especially software industry and a standard pathfinder non-existence in higher and academic education in universities, and lack of relation between industry and university, and also some technical reasons relating estimation method inefficiencies and inefficiency of using most of these methods in different companies and projects because of major factors and criteria non-conformity in different countries regarding local conditions and features of each region.

In continuation, these main challenges are addressed focusing upon technical problems, because technical problems to some extent can be removed by study and research in this domain.

4. Results Evaluation

To investigate more accurately major challenges in estimation it was essential to evaluate and study the most often used methods so that reasons of tendency toward using these methods can be specified and also their strength points be highlighted; moreover by investigating their weak points, reasons of broad deviations in estimated data and real data will be made clear. Also we'll address the reasons of limited use of famous and more often used methods such as Use Case Points, which is of the widest and the most often used methods present in the worldwide software industry, to recognize possible weak points of these methods and eventually provide solutions to minimize negative impacts.

4.1. Analogical Method

Analogical is the most often used method. In this method, estimation is done based on real experiences (data of real projects), therefore it prevents from recalling expert judgments and also algorithmic models complexities. Some formats can be made to show different features and characteristics of projects, so that differences between previous projects and the proposed project are expressly compared. Tools such as ANGEL, developed by England Bournemouth University, can be used [19]. This method is very simple to be applied to similar projects and estimation is rapidly done [20]. But from the amount of deviation observed in obtained results it can be concluded that this method has not correctly been applied to estimation and as merely compared with previous projects, the assessments are general and less accurate, and different factors and effective criteria in the field of development environment, technical issues and people's productivity and capability are not accurately investigated.

Moreover, in using this method there should be similar projects and their real data be available. Also despite these data availability, this method is dependent on expert judgment so that s/he computes the differences between previous projects and the current one by comparing them. So, it can be subjective, and the person's errors can have much impact on estimation. For example, two projects looking the same may function differently in a critical path. Moreover two analysts may doubt resemblances and differences and have different views of probable estimations, so in this method unreliability increases.

When the analyzer has no experience or data of similar projects, this method is not proper with expert judgment only. Therefore, this method is not useful when the suggested system differs from other projects in some directions. Also this method can be full of error when details have not been designed or when demands are not completely clear.

4.2. Expert Judgment Method

As it was evident the diagrams, expert judgment was among the most popular estimation methods in Iranian companies. Of course this is not because of capability and accuracy of this method, but it is mainly because of relatively inexpensiveness of expert judgment and no need to much familiarity with estimation methods. And also it is because of escaping from working on estimation and spending expense and time that many companies have practically selected this method. Of course the result of non-attention to estimation methods has also been specified in data conformity results.

Even if experts have direct experience in similar projects, although this method can have accuracy, given software industry condition in Iran this does not seem rationally and is somehow paradoxical.

Moreover, regarding needing no or limited previous data in this method, companies exempt themselves from gathering and identifying software metrics proportional to organization and staffs. That is, if the sense of need to gathering and keeping previous data in companies is made, besides increased estimation accuracy, it can significantly assist the organization with improving the process by developing software metrics.

The main defect of this method is that estimation is highly dependent on expert judgment, so it maybe subjective. Vision may be so limited in the process and factors the expert investigate to develop estimation. So making estimation and admitting it is difficult for the expert him/herself. Even documenting it is more difficult. This technique is problematic for the organizations which do not have software engineering expert group. When the experts use previous projects memories instead of previous database, there is limited evidence to evaluate estimation.

Even when it is clear that one project differs from another, it is not always evident that differences influence expense. A proper expense strategy from previous data is unreal in new project estimations because it is not necessary for new projects expenses to be linear: one cannot produce code twice with the same speed (Pfleeger 2005). There may be need to time to coordinate communications and to summarize differences in abilities, benefit and experience. For example, in classic software engineering study, ratio between the best and worst efficiency on productivity measurement average is 10 to 1, and relationship between experience and efficiency of programmers cannot be easily defined (Sackman 1968). In this method the level of knowledge is often questionable, and in many cases experts tend to be foundation [8].

Also, because of high degree of non-absoluteness in this method, this method is usually completed by other methods. This method can be useful when experts are dependent on experience and knowledge or when there is previous projects data and an understanding of the project proposed. If some aspects of the project to estimate are completely new to analyzer, this method should not be used. Expert judgment method is often used in project life cycle, especially before thoroughly specifying design of details or needs.

4.3. Use Case Points Method

Use Case Points (UCP) method is one of the most commonly used estimation methods in software industry and research fields. This method was provided by Gustav Karner in 1993 [24]. Since then it has been vastly given attention by researchers in industry and university. The main goal of this method was to provide an approach to estimate object-oriented projects. In Benta Anda study to evaluate the accuracy of this method in a wide international IT company, a comparison between estimation based on UCP method and an estimation done by 37 professional experts in 11 groups showed that this method has provided an estimation closer to real amount of project. The UCP criterion of MRE = 0.21 versus experts estimation method of MRE = 0.37 [25]. This method, if reformed, can be considered as a successful method to estimate software projects, but in current situation its capability has been reduced as it has some defects and its criteria and factors are not general in different organizations, that is factor and metrics available in this method are different in various companies.

Based on our separate study [29] done on this method the main problems of it are as follows:

- Lack of any distinct guide in writing use cases hence incorrectly accounting number of transactions
- Defect of weight measurement criterion of use cases which is the number of transactions
- General classification of use cases into three groups of simple, average, complex
- Difficulty of quantification of technical and environmental factors
- Changes in significance of some technical and environmental factors regarding software development in the recent decade
- People-hour factor in use case points (PHF in UCP) being dependent on peoples and conditions of companies which need consideration by them

Besides the problems investigated in methods used, there are more general problems in development process domain which are effective in making challenges. One of the main and most important challenges in all the process of software development and especially in project effort estimations is lack of attention to software engineering standards. None of the present estimation methods have been provided focusing on software engineering standards. In algorithmic methods which are mainly based on previous data of company and software metrics, when proposing criteria, these criteria have been qualitatively proposed. For example, one of the major metrics in some estimation methods is reliability (COCOMO, function points, etc.), but this metrics has been thoroughly proposed qualitatively and its quantification in each project is highly dependent on the expert's thoughts and experiences or the estimator, while in the presented standard to estimate ISO/IEC 9126 [26], the way of computing this criterion has been specified quite quantitatively, and the factors and formulae necessary to quantitatively investigate this criterion have been provided. So, there is a fundamental need to give attention to software engineering standards in all the processes of software development especially to estimation part.

The other problem is lack of familiarity with and use of CASE tools in software development process and as a result losing time and accuracy in this process which causes development teams to not give much attention to planning and management because of time limitation. For example Enterprise Architecture has been introduced as one of the important and successful tools of estimation in object-oriented development pathfinders which besides serving as a tool of analysis and design in a limited time can estimate the effort needed for the project.

Time limit in presenting the product to market is one of the other reasons that companies avoid spending time to provide an accurate estimation of the effort needed. This matter was proved in the previous section in selecting faster estimation methods, because most of the companies try to put the products into market as soon as possible, so they have less focus on management activities including estimation and also non-functional requirements which cause to waste resources and reduce quality of products in Iran.

5. Conclusion

In this study, effort estimation in Iranian software companies was addressed which was prepared based on the data collected from 32 software projects in some companies of different cities of Iran. According to this study it was made clear that expert estimation, analogical estimation, Use Case Points estimation and a combination of two methods of expert estimation and analogical estimation are the most often used methods in software projects. The obtained results showed that there was nearly more than 40% deviation in estimations. Reasons of these deviations were investigated and the most important causes of these deviations regarding strength and weak points of estimation methods were enumerated. Some general reasons relating estimation challenge in Iranian software companies were specified, remove or reduction of which can be effective in improving Iranian companies' software industry status. Finally, the main estimation challenges in Iran's software industry can be summarized as follows:

- Time limits of presenting the product to market
- Not having regular process in software development
- Lack of attention to software metrics in companies and not having metrics proportional to people's

capability and capacity and to the company's facilities for accurate estimation

- Not following software development standards
- Not learning and teaching in universities because of texts and resources being not up-to-date and of weak relationship between industry and university
- Not having CASE tools in software development process and lack of familiarity with tools
- Preliminary estimations do not have adjustment coefficient, simply speaking, estimations are not much reviewed.

References

[1] Sanjeev D, *Software Metrics – A Tool for Measuring Complexity*, International Journal of Software and Web Sciences s, 2 (1), pp. 4-7, 2012.

[2] Ahmad Z, Majeed S, *Towards Increase in Quality by Preprocessed Source Code and Measurement Analysis of Software Applications*, IST Transactions on Information Technology-Theory and Applications, Vol. 1, No. 1 (2),2010.

[3] Ahmad Z, *Towards performance measurment and Metrics Bassed Analysis of PLA Applications*, International Journal of Software Engineering & Applications (IJSEA), Vol.1, No.3, 2010.

[4] Galorath DD, Evans W. M, *Software Sizing, Estimation, and Risk Management*, Auerbach Publication (Taylor & Francis Group), 2006.

[5] Kashyap D, Tripathi A, Misra AK, *Software Development Effort and Cost Estimation: Neuro-Fuzzy Model*, IOSRJCE, 2012.

[6] Koteswara RK, Raju G.S.V.P, Madhusudhana RTV, *Effort Estimations Based on Lines of Code and Function Points in Software Project Management*, IJCSNS, Vol.8 No.6, 2008.

[7] Attarzadeh I, Hock OS, *Improving the Accuracy of Software Cost Estimation Model Based on New Fuzzy Logic Model*, World Applied Science Journal Vol.8 No.2, 2010.

[8] Prabhat KVB, Kumar MS, *Complexity Metric for Analogy Based Effort Estimation*, Journal of Theoretical and Applied Information Technology, 2009.

[9] Sharma A, Kushwaha DS, *Applying Requirement Based Complexity For The Estimation of Software Development and Testing Effort*, ITCS, SIP, JSE, 2012.

[10] Mohagheghi P, Anda B, Conradi R, *Effort Estimation of Use Cases for Incremental Large-Scale Software Development*, ICSE, 2005.

[11] Jorgensen M, Grimstad S, *Software Development Effort Estimation Demystifying and Expert Estimation*, Simula Research Laboratory, Norway, 2009.

[12] Reddy CH. S, Raju KVSVN, *An Improved Fuzzy Approach for COCOMO's Effort Estimation Using Gaussian Membership Function*, DENSE Research Group, Journal OF Software, Vol.4, No.5, 2009.

[13] Vijay FJ, Manoharan C, *Initial Hybrid Method for Software Effort estimation Benchmarking and Risk Assessment Using Design of Software*, Global Journal of Computer Science and Technology, 2009.

[14] Frohnhoff S, Engels G, *Revised Use Case Point Method -Effort Estimation in Development Projects for Business Applications*, Germany, 2008.

[15] Sharifloo Molzam A, Mirakhorli M, Shams Aliee F, *Agility in Iran: Position, Motivation, and Adaption*, 2008.

[16] Menzies T, Jalali O, Hihn J, Baker D, Lum K, *Software Effort Estimation and Conclusion Stability*, Vol.6, No.1, 2007.

[17] Poscic P, Pavlic M, Vrcek N, *Method for Estimating the Complexity of Designing Business Information Systems*, JIOS, Vol.32, No.2, 2008.

[18] Nicholson B, Sahay S, *Building Iran's Software Industry: An Assessment of Plans and Prospects*, EJISDDC, 2003.

[19] Sheppred M, Schofield C, *Estimating Software Project Effort Using Analogies. IEEE Transactions on Software Engineering*, vol. 23 no. 11, 1997.

[20] Kim S, Lively W, Simmons D, An *Effort Estimation by UML Points in the Early Stage of Software Development*, Software Engineering Research and Practice, 2006.

[21] Pfleeger SL, Felicia W, Lewis R, *Software Cost Estimation and Sizing Methods: Issues and Guidelines*, RAND Corporation, USA, 2005.

[22] Sackman H, Erikson WJ, Grant EE, *Exploratory Experimental Studies Comparing Online and Offline Programming Performance*, ACM Volume 11 Issue 1, 1968.

[23] Laird LM, Brennan MC, *Software Measurement and Estimation a Practical Approach*, wiley Interscience Publication & IEEE Computer Society, 2006.

[24] Karner G, *Resource Estimation for Objectory Projects*, 1993.

[25] Anda B, *Comparing Effort Estimates Based on Use Case Points with Expert Estimates*, 2003.

[26] ISO/IEC 9126-2, *Software Engineering, Product Quality, International Technical Report*, Part 2: External Metrics, JAPEN, *2002*.

[27] Humphrey W, *Winning with Software: An Executive Strategy, Addison* Wesley, 2002.

[28] Cohn, M, *Agile Estimating and Planning*, Pearson Education Inc, 2006.

[29] Yavari Y, Afsharchi M, Karami M, *Software Complexity Level Determination Using Software Effort Estimation Use Case Points Metrics*, Malaysian Conference in Software Engineering (MySEC), , IEEE Xplore , p 257-262, 2011.