THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

An Empirical Study on Software Development Challenges

Nader Saboktakin
Faculty Member and Ph.D. of Department of Knowledge and Information Science,
Faculty of Humanities, Persian Gulf University, Iran
Sedigheh Mohammad Esmaeil
Associate Professor, Department of Knowledge and Information Science,
Faculty of Humanities and Social Sciences, Islamic Azad University,
Science and Research Branch of Tehran, Iran

Abstract:
The software has been considered constantly by librarian as a new information tool and invention. Present research intends to study challenges of Custom computer software development in Iran and to submit appropriate solutions for challenges. According to Standish Group's Chaos Report, only few software is completed successfully while a considerable part of them are failed or faced with challenges. Taking time and costs more than initial estimation and quality less than initial expectations are regarded challenges. Studying resources, some of the most challenging factors in software development phases and areas which mentioned in scientific texts have been extracted and 201 questionnaires out of 234 questionnaires have been returned by managers of companies' project team i.e. more than 86%. Research variables had acceptable Cronbach's alpha, oscillating in range of 741% to 823%. Using mean test of a society, factors concerning process management and documentation quality have been recognized as challenges. Freidman text confirms significant difference, at least between the two factors, and challenges related to software maintenance has the lowest mean and the best situation while challenges related to documentation has the highest mean and the worst situation. Findings by Pearson correlation test offers positive and significant relation among all variables of research. The most important challenges for computer software developers in Iran are related to documentation quality and process management areas.

Keywords: *Software development, custom software, challenge, Iran*

1. Introduction

As links between information producing and consuming, not only librarian try to create ways for better arrangement in order to reach information recourses easier, but also they think about qualitative promotion of resources which present to target group due to librarians' work importance and their important role to provide information resources. Software is not only a highly innovative and economically important sector in its own right, but it is often also an important element of development in other sectors. Many process and structural development depend heavily on organizational changes that are facilitated by software innovations (Haryani and Gupta, 2016).

Software, particularly commercial (vs. Open Source) computer software (Sharifinia & Sepehr 2012), are focused intensely by librarians as a new informing tool and invention due to their undeniable abilities to save and restore large part of data, automatizing the affairs of libraries and information centers, promotion of security, accuracy and exactness for processes affected by their activities. Therefore, studying causes and factors in relation to computer software challenges can be effective for qualitative improvement and it may be used by librarians who work in the field of information, knowledge management and information science.

Computer software development principally is riskier compared with other industrial products. According to results by Standish group's chaos report that is well known to Software Crisis, 365 IT sections managers in 1994 have participated in study and they reported that only 16% of software development projects have been completed successfully, 53% have been faced with challenges and 31% have been failed (Marchewuka, 2012). A similar study in 2000 concerning the field shows that only 28% of software development projects have been completed successfully, 23% have been stopped and 49% have been faced with serious problems (Stepanak, 2005). Only 29% of software development projects have been completed successfully in 2004 while 53% & 18% has been challenged and failed respectively. Similarly, only 32% of software development projects have been completed successfully in 2009 and 44% & 24% has been challenged and failed respectively (Eveleens and Verhoef, 2010). Project success depends on suitable timetable of project completion in software development projects and this completion should depend on exact estimation of prime cost and software quality in development process and applying specifications and abilities which have been determined and predicted at the beginning of the project. Here, challenge means that a completed project imposes more cost and time and lower quality rather than initial expectations due to lack of appropriate planning and policy in time, cost and quality management. The project is regarded as failed if it stops in any phase of software life cycle or if it remains uncompleted.

Principally, several factors should be considered in software development. Software project failures are caused by internal and external causes (Xiagnan and et al., 2010; quoted in Lehtinen and et al., 2014). This research has been examined some of them. Parts of them are related to some phases (System Requirements, Programming Quality and Maintenance) of software development and the other parts are related to factors (documentation, personnel resources and process management) that their absence or failure can affect on development process.

Methodology consists of consecutive, successive methods and guidelines that are tools to formulate and regulate these processes with respect to evolutionary phases in software development. Hence, software developers use various methodologies such as Software Process Improvement (SPI), Agile (XP, Scrum, Kanban, etc.), and Rational Unified Process (RUP), Prescriptive or traditional and Specialized Process models in software manufacturing phases. Although, each mythology uses special techniques and standards in software development; however, they follow same principles and phases which observing them is avoidable. These principles begin with analysis, feasibility study and it leads to software manufacturing operation and finally maintenance and care after study on technical problems, planning, manufacturing & implementation, program test.

Regarding the fact that no such research has been done yet in the field of librarianship in Iran, necessity of exact attention in Iranian software industry and studying challenges and threats of this industry as a part of information and communication technology (ICT), improves librarian position as pioneers of information system. On the other hand, all efforts and activities in direction of software development and each phase or area is a mean to the end that the software will be able to remain for a long time after implementation and installation and to adopt changes, relative unavoidable, arisen from environmental conditions.

2. Literary Review

Taher, Ahmad and Kazirun (2010) studied challenges which companies encounter them to use standards and procedures in software development projects. They regarded amount of using methodology and standard programming by companies plus software failures correction in software development process. Results show that 60% of companies use methodology in software development constantly, so that, majority of those companies use agile methodology due to better attraction of customer's satisfaction, flexibility and appropriate speed to execute commercial projects. Also 77% of companies stated that they use coding standard for programming in projects and all software engineers who works in companies, have attained required educations to use coding standard. 19% of companies stated that they use sometimes coding standard. 4% of companies have not cared about coding standard and stated that they rarely use them in programming. Assurer group of software quality is responsible to examine and correct software failures and problems. 89% of companies have stated that they study and remove problems officially and periodically in every software project.

Kannabiran & Sankaran (2011) studied on factors in external software quality such as requirement uncertainty, trained personnel, maturity process, knowledge transfer and integration, communication and control, technical infrastructure. Results show that requirement uncertainty is extremely related to all aspects as for software quality, while trained personnel and maturity process are related to software quality to some average extent, meanwhile communication and control, knowledge transfer and integration, and technical infrastructure are related to software development quality less than others.

Shouki (2016) in a study discussed software projects still suffer from high failure rates, especially in developing countries. However, no one evaluated the current studies that were conducted in the context of Saudi Arabia (SA), a developing country. The findings of his study suggest that management/organization factors including lack of top management support, organizational culture, business process reengineering, lack of training, and unavailability of project management office are main factors that influence software project failures in SA. However, technical and financial factors are secondary; this might be due to the extensive government subsidization for IT. Our analysis can assist software project managers in SA, and it may apply to other developing countries in the Middle East.

3. Methodology

Present research which belongs to applied type is an analytical-survey research. It means that computer software development challenges have been studied by analytical-survey method. The questionnaire is also used by five-level Likert scale as a tool to gather information. Items of initial questionnaire have been produced and compiled in 6 dimensions and 30 items based on the most challenging factors reported by software developers and official valid resources have published these challenges in scientific texts. It was formulated as final questionnaire considering opinions and final confirmation by associate professor, assistant professor and software experts, and pilot survey in a society consist of 30 members of working people in valid ICT companies, then the questionnaire was distributed in research society by targeted method.

In other words, Library & Information Science and software engineer masters' opinions have been used in order to obtain questionnaire validity and the questionnaire has been confirmed by experts of this field. Also Cronbach's Alpha Coefficient was used for questionnaire reliability. According to alpha coefficient column in table-1; it is observed that research variables have acceptable alpha value.

Variables	Cronbach's Alpha
SYS	0.811
PGM	0.755
MA	0.823
DOC	0.861
PER	0.741
PM	0.799

Table 1: Cronbach's Alpha Coefficients for Research Variables

This research statistical society includes all computer software developers consisting also library software developers who have a valid technical confirmation certificate issued by Informatics Higher Council. According to Krejcie & Morgan Table of sample size of questionnaire; sample volume consists of 234 companies with ranks 1 to 7 for which questionnaires were sent. 201 questionnaires were filled and returned by project managers of software development team, therefore, 86% of questioners have been returned. Descriptive statistics and also inferential statistics including a society average tests, variance analysis and Friedman test were used to analyze data collected by gathered questionnaires, and Pearson coefficient test was used to examine the relation of research variables.

4. Results

4.1. What Are Demographical Specifications of Project Team Member?

This part of statistical analysis studies on way of statistical sample distribution obtained from computer software development companies regarding variables such as number of members of project team or group, duration of projects completion, average experience of project team members and used methodology in projects.

Number of Members of Project	Frequency	Frequency Percentage
1-2	14	70
3-5	70	34.8
6-10	56	27.9
11-20	35	17.4
More than 20	26	12.9
Total	201	100

Table 2: Distribution of Responders Regarding Number of Project Members

Table 2 of frequency distribution shows status of number of project team member. Among all sample groups, 14 project team among groups have 1-2 members (about 7%), 70 project team have 3-4 members (about 35%), 56 project team have 6-10 members (about 28%), 35 project team have 11-20 members (about 17%) and 26 project team have more than 20 members (about 13%) As it is observed in table 2, the highest frequency belongs to project team with 3-5 members with 34.8%.

Duration of Project Completion	Frequency	Frequency Percentage
Less than 6 months	40	19.9
6-12 months	73	36.3
13-24 months	51	25.4
More than 24 months	37	18.4
Total	201	100

Table 3- Distribution of Responders Regarding Duration of Projects Completion

Table-3 of frequency distribution shows status of duration of projects completion. Among all sample groups, 40 team have completed the project less than 6 months (about 20%), 73 team have completed the project between 6 to 12 months (about 36%), 51 team have completed the project between 13 to 24 months (about 25%) and 37 team have completed the project more than 24 months (about 18%). As it is observed in table-3, the highest frequency belongs to groups which have completed the projects between 6 to 12 months with 36.3%.

Average Experience of Team Members	Frequency	Frequency Percentage
Less than 1 year	1	0.5
1-3	47	23.4
4-6	109	54.2
7-9	34	16.9
More than 9	10	5
Total	201	100

Table 4: Distribution of Responders Regarding Average Experience of Project Team Members

Table 4 of frequency distribution shows status of average experience of project teams' members. Among sample groups, 1 team members are experienced less than 1 year (about 0.5%), 47 team members are experienced between 1-3 years (about 23%), 109 team members are experienced between 4-6 years (about 54%) 10 team between 7-9 years (about 17%) and 10 team members are experienced more than 9 years (about 5%). As it is observed in table 4, the highest frequency belongs to team in which members are experienced 4-6 years with 54.2%.

Used Methodology	Frequency	Frequency Percentage
No methodology	23	11.4
SPI[1] models	4	2
Agile	45	22.4
RUP	66	32.8
Other	63	31.3
Total	201	100

Table 5: Distribution of Responders Regarding Used Methodology in Software Development

Table 5 of frequency distribution shows status of used methodology of project in software development. 4 projects used SPI methodology (about 2%) 45 projects used Agile methodology (about 22%) 66 projects used RUP (about 33%) and 63 projects used other methodologies (about 31%). Also no methodology has been used by 23 projects. As it is observed in table 5, the highest frequency in used methodology in software development belongs to RUP methodology with 32.8%.

Table 6 shows descriptive statistical indices such as mean and standard deviation for all the questionnaire questions and research variables. That shows challenges mean in software development projects is relatively in average level and less than average.

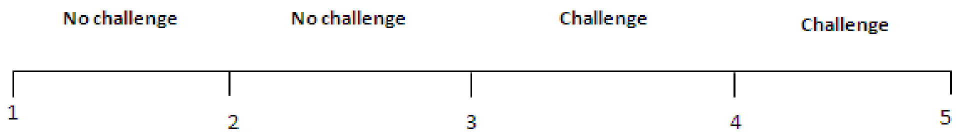
Also as it is observed in table 7 that mean of all challenges (system requirement, programming quality, maintenance, documentation quality, personnel resources and process management) is in range of 2.71 and 3.35.

4.2. What Are Challenges of Each Computer Software Development Phase and Area Including System Requirement, Programming Quality, Maintenance, Documentation Quality, Personnel Resources and Process Management?

Statistical hypothesis testing of a society i.e. one-sample test has been used in order to study status of challenges concerning computer software development that actually examines the difference between mean of the sample and a hypothesized value. Considering five-level Likert scale, null hypothesis in all research variables is in following form:

- H0: (null hypothesis) $\mu = 3$
- H1: (alternative hypothesis) $\mu \neq 3$

Considering the fact that all questionnaire questions have been designed directly, the mean in range of 3-5 means that the factor is challengeable and in range of 1-3 means there is no challenge. In other words, if mean of the factor related to personnel resources has a significant difference i.e. more than 3 (middle of range), then that factor is recognized as a challenge related to computer software and if the mean of the computer software development factor has no significant difference, i.e. it is less than 3 (middle of range), then it is not included as a challenge related to computer software. Following range shows conditions of results interpretation:



[1] - Software Processing Improvement

Indices	Challenges	Mean	Standard Deviation
1-System requirements have been recognized mistakenly	SYS	2.85	1.192
2- System requirements have been recognized incomplete or obscure		3.24	1.097
3- System requirements have been recognized unreal or conflict		2.5	1.082
4-Software quality requirements		2.89	1.101
5- System requirements are changed constantly		3.57	1.108
6-It is not conforming to programming standards		2.56	1.048
7-Comments are incomplete in relation to source code	PGM	3.25	1.127
8-Various modules are not allocated in program in a way that they be independent to each other concerning functionality and operating		2.75	1.105
9-The program is very complicated and restructuring is not possible		2.98	1.153
10- Inappropriate use of programming technique decreased ability of source code Comprehension		2.8	1.044
11-Submitted software systems are not comprehendible and analyzable easily	MA	2.76	1.001
12- Submitted software systems are not changeable and optimizable easily		2.87	1.041
13- Submitted software systems are not sustainable and resistible against unexpected effects arisen from changes		2.82	0.979
14- Submitted software systems are not testable easily		2.73	1.104
15-Overally, submitted software systems are not maintainable easily		2.68	1.019
16.The project has been documented unreliably or obscurely	DOC	3.21	1.107
17.There has been no documentation or they have been incomplete		3.26	1.134
18.Tracking previous documentation in project is hard concerning design specification & users' requirements		3.3	1.078
19-Changes have not been documented completely		3.62	0.983
20-Documentation are not consistent and comprehensive		3.38	1.018
21-Frequent replacements happen in project team	PER	2.99	1.17
22-Members of project team are not experienced or skilled sufficiently		2.78	1.046
23- Members of project team have not passed appropriate educations		3	1.077
24- Members of project team are not able to manage human resources & time		3.05	1.226
25- Members of project team are not obligated toward the project		2.56	1.139
26-There isn't managerial support and policies in software development process	PM	3.06	1.221
27-Project planning and control are not effective		3.01	1.065
28-There is no proper estimation of project execution schedule & cost		3.35	1.044
29-It is not effective to control changes in configuration management software		2.95	1.081
30- Quality control verifications are not effective to be sure from qualitative level		3.25	1.019

Table 6: Descriptive Statistical of Questionnaire Questions (Cont.)

Indices	Mean	Standard Deviation
SYS	3.0080	0.84305
PGM	2.8687	0.77889
MA	2.7721	0.78818
DOC	3.3552	0.85451
PER	2.8756	0.79464
PM	3.1264	0.81003

Table 7: Descriptive Statistic of Research Variables

Table 8 results show that factors related to documentation quality and process management have been recognized as challenge while factors related to system requirement, programming quality, maintenance and personnel resources are not considered as challenges because T statistic is less than 1.96 in all of them.

According to statistical hypothesis testing of a society, challenges related to documentation quality with T statistics calculated 5.894 is more than 1.96 and regarded as challenge. Also, challenges of process management with T statistics calculated 2.212 is more than 1.96 and regarded as challenge. Other challenges in relation to system requirement, programming quality, personnel resources and software maintenance is not considered challenges due to T statistics calculated respectively 0.134, -2.391, -2.219 and -4.099 are less than 1.96.

Indices	Significance Level	T-Statistic	Mean	Status
SYS	0.894	0.134	3.0080	No Challenge
PGM	0.018	-2.391	0.8687	No Challenge
MA	0.000	-4.099	2.7721	No Challenge
DOC	0.000	5.894	3.3552	Challenge
PER	0.028	-2.219	2.8756	No Challenge
PM	0.028	2.212	3.1264	Challenge

Table 8- Examining Status of Research Variables

4.2.1. Ranking Challenges as for Computer Software Development

Freidman test has been used to rank challenges of computer software development. Null hypothesis and one-hypothesis in a Freidman test are formulated as follows:

- H0: Means of ranks are equal
- H1: At least two ranks have different means

Present Status
Significance Level 0,000

Table 9: Results of Freidman Test

According to table 9 of Friedman test, it can be said that there is a significant difference at least between two factors because significance level is less than 0.05.

Research Variables	Rank Mean
SYS	3.39
PGM	3.23
MA	2.92
DOC	4.47
PER	3.09
PM	3.90

Table 10: Ranking Challenges as for Computer Software Development

As it is observed in table 10, challenges of software maintenance has the best status because they have the least rank of meaning while documentation quality and process management challenges have the worst status because they have the most rank of meaning. Consequently, challenges related to system requirement and programming quality has more mean of ranks with regard to each other respectively with 3.39 and 3.23.

4.2.2. What Are the Most Important Challenges for Computer Software Development Companies?

In order to determine the top 10 higher-severity problem factors of computer software development, the mean values of each problem factors were computed and stored in descending order.

Indices	Mean
1-Changes have not been documented completely	3.62
2- System requirements are changed constantly	3.57
3-Documentations are not comprehensive and consistent	3.38
4-There is no proper estimation of project execution schedule & cost	3.35
5-Tracking previous documentation in project is hard concerning design specification & users' requirements	3.30
6- There has been no documentation or they have been incomplete	3.26
7-Comments are incomplete in relation to source code	3.25
8- Quality control verifications are not effective to be sure from qualitative level	3.25
9- System requirements have been recognized incomplete or obscure	3.24
10- The project has been documented unreliably or obscurely	3.21

Table 11: The Most Challenging Cases Concerning Computer Software with Affiliated Means (Descending Order)

As it is observed in table 11, considering each question in relation to documentation quality, system requirements, programming quality, maintenance, personnel resources and process management; it can be said that the most challenging cases against computer software developers in software development are explained as follow:

As it is observed in table 11, items 1,3,5,6 and 10 are five important challenges related to documentation area for computer software developers, process management area (items 4, 8) and requirements phase (items 2 & 9) with two challenges, and programming phase (item 7) with one challenge are considered the most important challenges for software.

Lack of documentation of changes completely with mean of 3.62, lack of documentations comprehensiveness and consistency with 3.38, lack of possibility of previous documentation tracking with mean of 3.30, lack or incompleteness of documentation with mean of 3.26 and unreliable and obscure documentation with mean of 3.21 are the most important challenges of documentation.

Also lack of proper estimation of project execution schedule and cost with mean of 3.35 and inefficiency of quality control verifications with mean of 3.25 are the most important challenges related to process management. Finally system requirements constant change with mean of 3.57 and incomplete and obscure with mean of 3.24 are the most important challenges of system requirements and lack of inserting complete comments in source code with mean of 3.25 are the most important challenges of programming quality.

4.3. Research Hypothesis

There is significant relation between challenges of documentation quality (as the most challenging area) with challenges of software maintenance. Pearson test has been used regarding research question and to examine relation among research variables.

- H0: Correlation coefficient is null. (There isn't a significant relation between two variables)
- H1 : Correlation coefficient isn't null. (There is a significant relation between two variables)

If significance level is less than 0.05, null hypothesis is rejected and one-hypothesis is confirmed i.e. significant relation, and if significance level is more than 0.05, null hypothesis i.e. no relation is accepted and one-hypothesis is rejected.

Table 12 shows correlation among research variables. As results and significant level shows, there are significant relations among research variables. Therefore, regarding calculated correlation coefficient of 0.34, there is significant relation between documentation challenges and maintenance challenges. Also there is a significant relation between process management challenges (as the next most challenging area) with maintenance phase challenges regarding calculated correlation coefficient of 0.28, therefore null hypothesis is rejected. These linear relations are positive. It means more challenges in areas of documentation and process management increase maintenance challenges and similarly, less challenges decrease software maintenance challenges.

Problem Factors	Correlation Coefficient (Significant Level)					
	(1)	(2)	(3)	(4)	(5)	(6)
(1) System requirement problems	1.00					
(2) Programming quality problems	0.59**	1.00				
	(0.00)					
(3) Software maintenance problems	0. 2**	0.29**	1.00			
	(0.005)	(0.00)				
(4) Documentation Quality problems	0.5**	0.42**	0.34**	1.00		
	(0.00)	(0,00)	(0.00)			
(5) Personnel resources problems	0.65**	0.56**	0.14**	0.43**	1.00	
	(0.00)	(0.00)	(0.008)	(0.00)		
(6) process management problems	0.62**	0.61**	0.28**	0.5**	0.61**	1.00
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	

Table 12: Pearson Correlation Coefficient to Examine Relation among Research Variables
***Correlation Is Significant at the 0.01 Level (2-Tailed)*

5. Discussions and Conclusions

Considering librarians' concern as pioneers to facilitate reach of information and their hesitancy for necessity of information resources for target groups, in addition to increasing abilities and capacities of computer software, it is unavoidable to study difficulties which computer software developers would face them.

Principally, several internal and external causes should be considered in software failures. This research has been examined some of them. Parts of them are related to some phases (System Requirements, Programming Quality and Maintenance) of software development and the other parts are related to factors (documentation, personnel resources and process management) that their absence or failure can affect on development process.

As regard to demographical specifications; the highest frequency of number of project group members belongs to team with 3 to 5 (35%), also, computer software projects have been completed often in period of 6-12 months (36.3%). More than half the software developers' team have experiences 4 to 6 years (54%) and they often have used RUP methodology (32.8%) in software development. This is in contrary to Taher, Ahmad and Kazirun (2010) results that show majority of companies use agile methodology due to better attraction of customer's satisfaction, flexibility and appropriate speed to execute commercial projects.

Considering statistical hypothesis testing, means of challenges concerning documentation quality and process management have been recognized as challenges in computer software development but other areas and phases have not been specified as challenges. Also taking Friedman test into account and ranking the computer software development challenges, challenges related to documentation quality have the highest rank mean and the worst status, subsequently; process management challenges have the highest rank mean and the worst status too, while challenges related to software maintenance have the lowest mean and the best status. Also, there is significant relation between documentation & process management with software maintenance phase based on research hypothesis and using Pearson correlation test. Therefore to overcome challenges of maintenance phase in computer software development in Iran and presenting appropriate solutions, lack of documentation is the most important challenges for computer software developers. Also based on results part of challenges of maintenance phase is related to challenges of Process Management.

To reduce or remove challenges related to documentation in country, Software developers should provide a set of documents that show accomplishment and evolution of their activity from the beginning to the end of project, so all the members of project team are obligated to document reasons of choosing their methods and approaches in all steps and procedures of software development clearly and they should document strategies and solutions to overcome problems and challenges completely.

Another challenge of software development is related to process management. To eliminate or reduce challenges related to process management in country there should be accurate estimation of costs and schedules of projects and quality control verifications should be effective to assure for qualitative level of projects.

6. References

i. Collin, S.M.H. (2004). Dictionary of ICT. London: Bloomsbury.

ii. Emadi, S. (2009). Presenting a model to study executive ability of software architecture, (PhD dissertation), Islamic Azad University, Science & Research Branch, Tehran, P.14.

iii. Eveleens J. L.; Verhoef, C. (2010), The Rise and fall of the Chaos Report Figures. IEEE Computer Society, IEEE Software, Vol. 27, no. 1, January and February, 30-36.

iv. Haryani, Sh. Gupta v.b (2016). Factors Affecting Innovation Capability Of Indian Software Firms (With Special Reference To Indore City). International Research Journal of Engineering and Technology (IRJET). Volume: 03 Issue: 12.

v. Informatics Higher Council. (2004), Second phase of software development standards & engineering system: Project management plan standard, State planning & management org. Tehran, P 7.

vi. Kannabiran, G. & Sankaran, K. (2011), "Determinants of software quality in offshore development. An empirical study of an Indian vender", Information and software technology, 53, 1199-1208.

vii. Lehtinen, T. O.A.; Mantyla, M. V.; Vanhanen, J; Itkonen, J; Lassenius, C. (2014), "Percieved causes of software project failures – An analysis of their relationships", Information and software technology 56, 623-643.

viii. Marchewka, Jack T. (2012), Information Technology Project Management, New York, John Wiley and Son. pp.5, 18.

ix. Pigoski, T. M. (1994). Encyclopedia of software engineering. New York, John Wiley and Son. Vol. 1, P. 619.

x. Sharifinia, A. A. & Sepehr, F. (2012), Recognition of Open Source Softwares. Tehran, Ketabdar.

xi. Shouki A. Ebad (2016). Influencing Factors for IT Software Project Failures in Developing Countries — A Critical Literature Survey. Journal of Software. Volume 11, Number 11.

xii. Stepanek, George. (2005), Software project secrets: Why software projects fail, Berkeley, Apress .PP.3, 4.

xiii. Tahir, A., Ahmad, R. & Kasiran, Z. M. (2010), An Empirical Study on the Use of Standards and Procedures in Software Development Projects, International Conference on Software Technology and Engineering, 2nd. Vol.1, 274 - 278.

xiv. Xiangnan, L. Hong, Y. Weijie, (2010), "Analysis failure factors for small and medium software projects based on PLS method", in: The 2nd IEEE International Conference on Information management and Engineering (ICIME), PP. 676-680.

xv. Yip S.W.I. & Lam, T. (1994), "A Software Maintenance Survey", Software Engineering conference, proceedings, First, Asia-Pacific, pp. 70-79.