ISSN No:-2456-2165

Computer Software Development Challenges in Iran, with Respect to Software Metrics

Nader Saboktakin
PhD of Department of Knowledge and Information Science, Persian Gulf University, Bushehr, Iran
Sedigheh Mohammad Esmail
Associate Professor of Department of Department of Knowledge and Information Science,
Azad University of Science and Research Branch of Tehran, Iran

Abstract- The aim of this research is to survey the challenges of computer software developers' companies in country, and based on research findings to examine software metrics. By literary review, some of the most problem factors in the computer software development published in scientific context were obtained. The research population consists of about 600 computer software developers' companies. 201 questionnaires out of 234, amounting to %86, were returned by project managers. Resulting of two sample tests with significance level greater than 0.05, there is no significant difference between the number of members of software project teams and variables related to the challenges of documentation qualities, system requirements, programming quality, maintenance, personnel resources and process management. The significant relationship between software project duration and system requirements were observed and the companies that run the projects shorter in duration, having more challenging to requirements. There are more challenges related to the qualities of software programming and maintenance in groups whose members are less experienced. Significant difference between the various methodologies adopted in software development with the above mentioned variables were not observed.

Keywords:- *Computer Software; Software Development; Challenges; Iran; Software metrics.*

I. INTRODUCTION

Computer software development principally is more risky 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 [1]. 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 [2]. 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 [3]. 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 which have been determined and predicted at the beginning of the project. So, 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 [4]. Software development, the ability to produce novel and useful software systems, is an important capability for software development organizations and information system developers alike. However, the software development literature has traditionally focused on automation and efficiency while the innovation literature has given relatively little consideration to the software development context. As a result, there is a gap in our understanding of how software product and process innovation can be managed. Specifically, little attention has been directed toward synthesizing prior learning or providing an integrative perspective on the key concepts and focus of software innovation research [5]. Principally, several factors should be considered in computer software development that parts of them are related to software development phases and accompanied metrics which are used to assure the quality of products in such a way that their absence or failure can affect on computer software development. Software development process means a part which is related to all phases of software development and its goal is to manage software life cycle in order to supervise and evaluate better and finally, to decrease challenges and promote qualities. Procedure of how to do something well, or Methodology also consists of consecutive, successive methods and guidelines that are tools to formulate and regulate these processes. Hence, software developers use various methodologies such as Software Process Improvement (SPI), Agile and Rational Unified Process (RUP) 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 unavoidable. These principles begin with analysis, feasibility study and it leads to software manufacturing operation and finally maintenance after study on planning, manufacturing & implementation, program test. Some reasons of the challenges, failure or success of software projects are made because of comprehensive attitude to observe these phases; focusing components such as time, cost, quality and goal achievement. Some reasons of the challenges, failure or success of software projects are made because of comprehensive attitude to observe these phases; focusing components such as time, cost, quality and goal achievement. On the other hand, studying abilities and level of manpower proficiency which mainly include project group members and particularly project managers and programmers is considered as factors may affect on success or failure of projects concerning computer software development.

By the way all efforts and activities in direction of software development and each phase are means 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. Lack of adaption with new conditions by the software makes it less usable and it will be out of date or retired after a while. One example is disk operating system (DOS) that is out of date at present time.

We deal with measurement and evaluation in our life every day. For example; when we go shopping, we buy things based on our incomes and/or we stirred sugar into our tea based on our previous knowledge and experiences.

Number of Members of Project	Frequency	Frequency Percentage	Cumulative percentage
1-2	14	7	7
3-5	70	34.8	41.8
6-10	56	27.9	69.7
11-20	35	17.4	87.1
More than 20	26	12.9	100
Total	201	100	-

Table 1. Distribution of responders regarding number of project members

According to one approach, any kind of study executed in the field of software by mathematics and statistics may be called “software metrics” due to the fact that quantitative analysis finally improves processes and procedures in and makes optimal use of resources and promote software quality. Also, using software metrics, software developers would be able to have more accurate predictions of costs and period of project execution and allocation of sources, therefore, it promotes productivity for software development. Hence, the software metrics intends to present indices for managers of software developer companies to evaluate software qualitatively by quantitative statistics and measurements. Organization managers in different levels such as top, middle and low-level managers i.e. project managers can gain useful information in the field of products quality by software metrics. On the other side, software developers would be able to evaluate final product better by quantitative information in relation to processes and procedures which they achieve during software development and its various phases, thus, software metrics is a way to solve the software development challenges. It should be noted that merely a part of principles and rules of software metrics are in direct relation with phases of software development and they are applicable in all the software development phases and the other part is related to software development indirectly including items such as manpower measurement and examining their proficiency and skill at the beginning of the project in order to avoid damage and loss arisen from such items. Although initially the very intention of this paper was to reveal some problem factors of companies' computer software developers across the country, but as a consequence, obtaining some hints to software metrics is also remarkable. As quantitative measurements are significant for all science, so is the case for computer science as well. In general what is basically meant by metrics.

Duration of Projects Completion	Frequency	Frequency Percentage	Cumulative percentage
Less than 6 months	40	19.9	19.9
6-12 months	73	36.3	56.2
13-24 months	51	25.4	81.6
More than 24 months	37	18.4	100
Total	201	100	-

Table 2. Distribution of responders regarding duration of projects completion

Including software metrics, is to regulate some rules and principles to assure software and papering the ground for future better quality. To specify software metrics, Software measurement provides continuous measures for the software development process and its related products. It defines, collects and analyzed the data of measureable process, through which it facilitates the understanding, evaluating, controlling and improving the software product procedure [6]. By the way software metrics divided into three categories including product, procedure and process metrics. One of the types of software metrics is Process metrics, and it means measuring the process of software development including human resources, time, etc.

II. RESEARCH QUESTIONS

- What are demographical specifications of teams’ members with respect to software metrics.
- Is there a significant relation between demographical specifications including project groups’ members, period of projects completion, average experience of project

groups' members and the used methodology in projects with related variables to computer software development challenges.

III. 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.

Thus, 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, particularly in [7].

Average Experience of Team Members	Frequency	Frequency Percentage	Cumulative percentage
Less than 1 year	1	0.5	0.5
1-3	47	23.4	23.9
4-6	109	54.2	78.1
7-9	34	16.9	95
More than 9	10	5	100
Total	201	100	

Table 3. Distribution of responders regarding average experience of project groups members

In other words, 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 and it was about 0.8. As it is observed research variables have acceptable alpha value. It was formulated as final questionnaire considering opinions and final confirmation by 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. This research statistical society includes 600 computer software developers who have a valid technical confirmation certificate. 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, about 86% of questioners have been returned. Descriptive statistics and also inferential statistics including analysis variance tests were used to analyze data collected by gathered questionnaires.

IV. RESULTS AND DISCUSSIONS

Research results are presented in this part regarding research questions.

What are demographical specifications of project teams' members with respect to software metrics?

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 projects' teams or groups, duration of projects completion, average experience of project teams' members and used methodology in projects.

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

Used Methodology	Frequency	Frequency Percentage
SPI models	4	2
Agile	45	22.4
RUP	66	32.8
Other	63	31.4
No methodology	23	11.4
Total	201	100

Table 4. Distribution of responders regarding used methodology in software development

We can also obtain some hints of software metrics which called basic or primarily metrics [8]. The basic reason why developers and managers need to use software metrics is that they can assess the quantity (and quality) of software artifact more easily, quantity relates to the size and complexity of the software and may be useful in predicting the resources i.e. developers, time, etc. required to complete or maintain a piece of software [9].

The less number of members of projects, the less magnitude and complexity of the projects and consequently need to use less resources will be. So about 7% projects completed by 1-2 members and about 70% of them completed by less than 11 members. An indication of using less resources in comparison with software development projects which completed by many members.

Table 2 of frequency distribution shows status of duration of projects completion. Among all sample groups, 40 groups have completed the project less than 6 months (about 20%), 73 groups have completed the project between 6 to 12 months (about 36%), 51 groups have completed the project between 13 to 24 months (about 25%) and 37 groups have completed the project more than 24 months (about 18%). As it is observed in table 2, the highest frequency belongs to groups which have completed the projects between 6 to 12 months with 36.3%.

Also applying basic metrics, the less time consuming the projects, the less magnitude and complexity of the projects and consequently need to use less resources will be. So about20% projects spend less than 6 months and about 56% of them completed less than 1 years. Showing that using less resources in comparison with software development projects which completed in longer duration. For measuring Schedule Slippage as a problem factor of software development, it can be calculated by getting (actual number of days subtract estimated number of days) to be divided (estimated number of days) multiply 100 per project.

Research Variables	DOC	SYS	PGM	MA	PER	PM
Significance Level	0.397	0.819	0.573	0.093	0.339	0.828

Table 5. Results of mean test of two societies regarding number of project team members

Research Variables	DOC	SYS	PGM	MA	PER	PM
1-2	3.6143	2.9000	2.9857	2.9714	2.6000	3.2286
3-5	3.4657	3.0000	2.7571	2.9286	2.7886	3.0686
6-10	3.2500	3.0679	2.9714	2.6679	2.9071	3.1071
11-20	3.2343	3.0800	2.9143	2.5314	2.9714	3.1200
More than 20	3.3077	2.8615	2.8231	2.7923	3.0615	3.2769

Table 6. Research variables mean regarding number of project team members

Table 3 of frequency distribution shows status of average experience of project teams' members. Among sample groups, 1 group members are experienced less than 1 year (about 0.5%), 47 groups members are experienced between 1-3 years (about 23%) 109 groups members are experienced between 46 years (about 54%) 10 groups between 7-9 years (about 17%) and 10 groups members are experienced more than 9 years (about 5%) As it is observed in table 3, the highest frequency belongs to groups in which members are experienced 4-6 years with 54.2%.

Because there are many types of software metrics, to choose and use an appropriate and correct methods and tools related to them. They have close relationship with the extent of experience of project members. So it assumes that those project members whom have more experience, possibly can choose and use better of benefits of software metrics.

Table 4 of frequency distribution shows status of used methodology of project in software development. 4 projects used SPI methodology (2%), 45 projects used Agile methodology (about 22%), 66 projects used RUP (about 32%) Also no methodology has been used by 23 projects. As it is observed in table 4, the highest frequency in used methodology in software development belongs to RUP methodology with 32.8%.

Software metrics are recognized as a required tool in the big picture of software process improvement. So Using SPI in software development projects is an indication of applying software metrics in advance. As it can be seen only 4 projects have used SPI models.

Research Variables	DOC	SYS	PGM	MA	PER	PM
Significance Level	0.550	0.046	0.652	0.656	0.993	0.489

Table 7. Results of mean test of two societies regarding period of project completion

Research Variables	DOC	SYS	PGM	MA	PER	PM
Less than 6 months	3.5200	3,200	2.9550	2.8550	2.8750	3.1100
6-12 months	3.2959	3.0904	2.9123	2.8055	2.8795	3.2301
13-24 months	3.2941	2.9647	2.8157	2.6588	2.8941	3.0941
More than 24 months	3.3784	2.6973	2.7622	2.7730	2.8432	2.9838

Table 8. Research variables mean regarding number of period of project completion

To explain the importance of SPI, if an organization misunderstood initial requirements it forces to rework many of the implemented features. The delay of new features development takes for over 40% of the planned time. As a result of SPI, a requirements tracking method was implemented at this organization using a simple template for describing requirements , and this changes reduce the new feature development delay to only 10% of the planned time[10].

2- Is there significant relation between demographical specifications including project team members, duration of projects completion, average experience of project team members and used methodology in projects on the one hand with related variables of computer software development challenges including documentation qualities (DOC), system requirements (SYS), programming quality (PGM), maintenance (MA), personnel resources (PER) and process management (PM) on the other hand?

According to table 5 and analysis of performed variance; it can be said that computer software challenges in groups with various members have no significant difference. Level of significant difference is determined by significance value. If the significant value will be more than 0.05, null hypothesis is accepted i.e. means are equal.

Research Variables	DOC	SYS	PGM	MA	PER	PM
Significance Level	0.320	0.104	0.008	0.049	0.260	0.056

Table 9. Results of mean test of two societies regarding average experience of project team members

Research Variables	DOC	SYS	PGM	MA	PER	PM
Less than 1 year	3.8000	3.2000	3.6000	2.0000	2.2000	3.4000
1-3 years	3.5787	3.2723	3.2170	2.9830	3.0213	3.3872
4-6 years	3.2734	2.9248	2.7706	2.7541	2.8220	3.0037
7-9 years	3.3176	3.0118	2.7294	2.4941	2.9706	3.2294
More than 9 years	3.3800	2.6400	2.7000	3.0000	2.5200	2.8600

Table 10. Research variables mean regarding average experience of project team members

Table 6 shows variables mean regarding number of various members of project team in software development. As it is observed, challenges concerning computer software development are rather equal in different projects with various members' quantity participating in software development. Comparing mean of research variables based on period of projects completion.

According to table 7 and analysis of performed variance; it can be said that, only challenges related to system requirement in groups with different period of completion has significant difference among challenges of computer software development, but other challenges in groups with different period of completion have no significant difference.

Table 8 shows variable mean regarding period of projects completion in various times. Challenges related to system requirements are more in groups in which projects have been done less than 6 months. Generally, shorter period of projects completion leads to more challenges in system requirements challenge. Comparing mean of research variables based on experience of projects team members.

According to table 9 and analysis of performed variance; it can be said that, challenges related to programming quality and maintenance in groups with different average experience of members have significant relation among challenges of computer software development, but other challenges in groups with average experience of members have no significant difference.

Research Variables	DOC	SYS	PGM	MA	PER	PM
Significance Level	0.510	0.425	0.821	0.267	0.657	0.208

Table 11. Results of mean test of two societies regarding used methodology in software development

Table 10 shows variables regarding average experience of project teams' member. Challenges related to programming quality are more in team with less experienced members. In other words, less experienced members in project team leads more challenge in programming quality. So we can obtain some hints from software metrics, because one of the most challenges phases which affect on quality of artifact and the correct estimation of expected time and costs is programming. So using more experienced member in software project teams can lead to release software with better quality and maintainability. Comparing mean of research variables based on used methodology in software development. Comparing research variables mean regarding used methodology. According to table 11 and analysis of performed variance; it can be said that, challenges concerning computer software development in various projects with different methodology have no significant difference.

Table 12 shows variables mean regarding different methodologies used in software development. Computer software development challenges are relatively equal with different methodologies in various projects of software development. As it is observed, some challenges concerning computer software development including challenges of system requirement and also programming quality and maintenance have significant relations respectively with variables of demographical specifications including period of projects completion and average experience of project group members. But they don't have significant relations with other variables of demographical specifications including number of group members and used methodology in projects. Therefore, it is recommended to computer software developers companies to ask more time when they conclude contract with employers in order to remove and/or decrease challenges of computer software development for software projects completion. Also, they should use experienced experts and programmers in order to promote programming quality in computer software development. This leads to increase maintainability of software.

Research Variables	DOC	SYS	PGM	MA	PER	PM
No use	3.6348	3.2174	2.9826	3.0783	2.9739	3.3652
SPI Models	3.4000	3.1000	2.5500	3.1500	2.7500	2.6000
Agile	3.3867	2.9333	2.8844	2.7600	2.8489	2.9778
RUP	3.2697	3.0970	2.8152	2.7030	2.9667	3.1000
Other	3.3175	2.8857	2.8921	2.7175	2.7714	3.2063

Table 12. Research changes mean regarding mean of used methodology in software development

V. CONCLUSION

According to findings the most impediments that computer software developers are faced across the country related to challenges of system requirement and also programming quality and maintenance respectively with variables of demographical specifications including period of projects completion and average experience of project group members. To remove or reduce software challenges, the complexity of the software should be reduced firstly. Because each challenges embedded some sorts of complexity which can be resolved by providing certain solutions. So software metrics are examined not only to reveal these challenges but also they provide some hints to resolve them. As a result of taking advantage of software metrics the quality of software improve and there is better control over estimated cost and time to obtain.

REFERENCES

[1]. Marchewka, J. T. (2012). Information Technology Project Management. New York: John Wiley and Son.

[2]. Stepanek, G. (2005). Software project secrets: Why software projects fail. Berkeley: Apress .PP.3, 4.

[3]. Eveleens J. L.; Verhoef, Ch. (2010). The Rise and fall of the Chaos Report Figures.IEEE Computer Society.

[4]. Mensah, E. (2003). Software Development Failures: Anatomy of Abandoned Projects. USA: Massachusetts Institute of Technology.

[5]. Rose J. and Furneaux, B. (2016). “Innovation Drivers and Outputs for Software Firms: Literature Review and Concept Development,” Advances in Software Engineering, vol. 2016.

[6]. Honglei, T.,Wei,S.&Yanan,Zh. 2009. The Research on Software Metrics and Software Complexity Metrics.International Forum on Computer ScienceTechnology and Applications. China, IEEE.

[7]. Chen, J. Ch.& Huang, S.J. (2009). An Empirical Analysis of the Impact of Software Development Problem Factors on Software Maintainability. The Journal of Systems and Software, 82, 981-992.

[8]. Baker, M. D. (1991). Implementing an initial software metrics program.Modern technology corporation, Ohio, pp. 1289-1294.

[9]. [9]. Gunnalan,R., Shereshevsky, M &Ammar, H.H. (2005). Pseudo Dynamic Metrics.Lane Department of computer Science, Verginia, pp. 1-4.

[10]. [10]. Hadad, H. M. & Meredith, D. E. (2011). Instituting Software Metrics in Small Organizations: A Practical Approach. Eighth International Conference on Information Technology: New Generation. IEEE Press, pp. 227-232.