

Available online at <http://www.mecs-press.net/ijeme>

## Management of Changes in Software Requirements during Development Phases

Mohammad D. Aljohani <sup>a</sup>, M. Rizwan J. Qureshi <sup>b</sup>

<sup>a,b</sup> MS student, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

---

### Abstract

Change, in software requirements during its development phases, is considered as a major risk which may affect a software project to fail. Therefore, software engineering processes, methods, and tools are used in order to manage these risks whereas changes in requirements have many negative influences such as effects on system development life cycle (SDLC) phases, project progress, and outcome of a software project. Consequently, project managers must use risk management strategies, activities, and estimation techniques in order to manage and mitigate these risks which are caused due to changes in requirements. A novel model is proposed in this paper to manage and mitigate risks related to changing requirements. The proposed model is validated through qualitative research design. The results are in favor of the proposed model to show its effectiveness. It is anticipated that the proposed model will solve the problems of software companies in major to deal with risks about changing requirements.

**Index Terms:** Change requirement risk, Risk management, Risk mitigation, Risk estimation, Cost Estimation Techniques.

© 2016 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

---

### 1. Introduction

The software development projects in general are not prone of risks. Therefore, it is important to mitigate, monitor and manage risks to deliver software project as per the estimated cost and scheduled with high quality. It is difficult to identify and manage risks during software projects. The managers need to know the probability, impact of risks, costs and efforts to manage risks to complete projects successfully. The impact of software risks, in early phases of software development, is the highest [1]. The projects risks are divided into three types, i.e.; project, technical and business [2]. Whereas, Project risk: those risks that affect directly on the management of the project such as change requirement, budgetary and scheduling risks. In addition, Technical

\* Corresponding author.  
E-mail address: [rmuhammad@kau.edu.sa](mailto:rmuhammad@kau.edu.sa)

risk: those risks that related to the development and technical aspects of the project for example interfacing, maintenance, and implementation risks. Moreover, Business risk: those risks that can impinge on a project which derives from customer environment. For example changes in policy in the customer department, changes in customer organisation and changing priorities of the client. The project manager needs to know as many as possible the potential risk to the software, then analyze and assess its seriousness with estimation techniques. After that the strategies of dealing with that risk, planning, control and monitoring are used [2].

In this paper, we will discuss the major risks and propose a new model to mitigate and manage risks. The project managers must estimate the resources, effort, time and cost of risks to complete software projects successfully. There are several cost estimation methods as shown in fig. 1 [3]. The objective of this paper is to address the known limitations of the existing cost estimation methods by proposing a novel model to improve the accuracy of risks assessments.

The paper is organized as follows. Section 2 outlines the related work. Section 3 describes the problem statement. Section 4 depicts the details of the proposed solution. Section 5 contains the discussion and section 6 provides the validation of proposed solution.

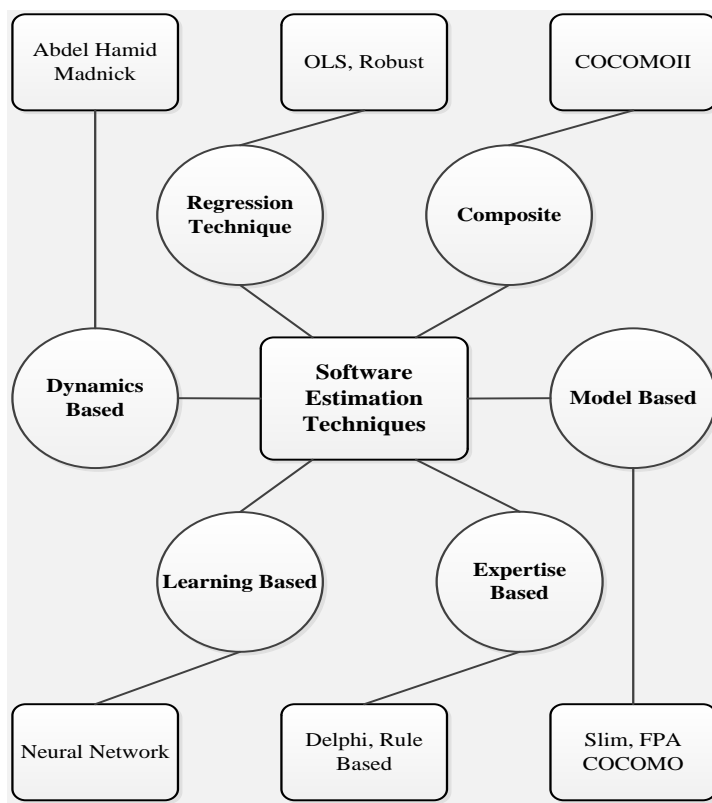


Fig.1. Software Estimation Techniques.

## 2. Related Work

The project managers are always willing to identify the potential risks those may lead to fail projects. As soon as a risk is identified, it is analyzed and assessed to judge its seriousness. A risk mitigation, monitoring and management plan is prepared to deal with it [2]. There are many risk management models available. The

most widely used model is introduced by the Software Engineering Institute (SEI) as shown in fig. 2. The Software management model has six processes, i.e., identify, analyze, plan, track, control and communicate risks [4]. In order to manage the risk properly, the manager should identify risks those can directly affect the project. In addition, planning turns risk information into decisions and actions. It is important to track and monitor risks. Contingency actions are planned to execute if the risks are going to affect a project.

Software risks directly affect to schedule and costs of projects. Hijazi et al. [5] provide a list of risks that software companies are facing during system development life cycle (SDLC). Falahah [6] proposes risk assessment model using (SERIM) method. A case study is implemented to develop an information system naming TrainSys to test the proposed model. The SERIM method used probability approach to focus on technical, cost and schedule risks. The requirements and complexity issues are not considered in the proposed model.

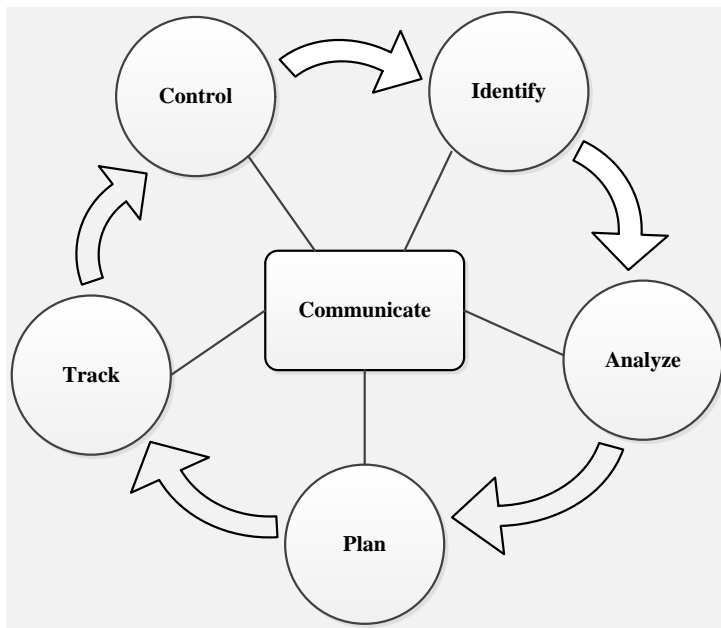


Fig.2. Software Risk Management Activities.

Alqrainy and hijazi [7] propose risk management strategies in the analysis phase. The major risk factors are identified in software projects, and a set of strategies are proposed to mitigate and manage the risks those are raised in the analysis phase. Alqrainy and hijazi [7] do not cover that how to estimate risks in the design and implementation phases of system development life cycle (SDLC).

Rout et al. [8] propose a mitigation model to manage risks throughout the SDLC. Chowdhury and Arefeen [4] discuss risk mitigation and assessment techniques. The classification of software cost estimation models and techniques are reported [3]. The existing software cost and effort estimations methods are Analogy, Expert Opinion, Putnam's Software Life-cycle Model (SLIM) and Constructive Cost Model (COCOMO). The comparative analysis of software estimation techniques and the aspects of each method are discussed. Analogy based estimation needs historical data that is not available in the majority of software development companies. Delphi estimation method requires an expert opinion on project specification document. Putnam is used to estimating the relationship of effort and people as a function of delivery time. COCOMO needs the accuracy of effort adjustment factors (EAF) to calculate the cost. Suri and Ranjan [3] applied a case study to estimate risks on a module. The results of the case study need to be generalized by applying it in other settings.

Software risks are analyzed and managed at an early stage of software development to avoid project failure [9]. The risk management activities are reported. It consists of various activities associated with software development to manage and control risks [9]. A novel methodology (Fuzzy-Excom) is proposed, combining Fuzzy technique features and COCOMO, to estimate risks. The proposed methodology is tested using three industrial data sets to prove that how much it is effective to assess risks of software projects. The result shows that there is a high correlation between size and effort for all data sets. Therefore, it is proved that Fuzzy-Excom is better in quality to estimate risks as compared to original Expert COCOMO. On the other hand, the future investigation is needed in order to improve the accuracy and sensitivity of proposed model by considering the features of other methods in risk assessment such as neuro-fussy logic.

Kaushik et al. [10] propose a risk estimation model using COCOMO based artificial neural network to predict software effort from perception network. The model is implemented on Matlab and COCOMO data sets are used to train and test the network. Data sets are divided into two subsets to achieve high accuracy, i.e., training and validation.

The generic model uses Principal Component Analysis (PCA) with Artificial Neural Network (ANN) model to improve the accuracy of cost estimation [11]. The combination of these models aids to estimate software cost and effort by calculating the size and create a hybrid tool. The objective is to propose a model that can be used to have more accurate estimations.

Dan [12] improved the accuracy of software effort estimation by proposing PSO-ANN-COCOMO II model. The original ANN-COCOMO II model can solve the problems of vagueness and uncertainty of the software attributes, but it still has some shortcomings such as the low speed of convergence and the high dependency on the initial weights. The PSO-ANN-COCOMO II model showed better results than the ANN-COCOMO II model.

Nagar and Dixit [13] proposed an effort estimation model with multi stages. The effort is estimated many times during SDLC. The results are compared to measure the differences. Basri et al. [14] propose a novel effort estimation approach to managing changes in requirements.

Saroaha and Sahu [15] reviewed tools and methods of software effort estimation based on use case point (UCP) model. The objective is to show that UCP model is more effective and efficient as compared to other models such as COCOMO and FPA. It is difficult to use UCP model because there is no standard to apply use case model.

There are many problems associated to change in requirements during the development of projects such as extra cost, more complexity, and probability of errors [16]. Therefore, a change management model is proposed to analyze and determine the relationship between existing and newly suggested requirements [16]. The proposal uses descriptive statistics and Spearman's rho. In the former, the changing requirements in the software development life cycle are looked at the five SDLC phases namely requirement analysis, design, implementation, testing, and maintenance. The results of the descriptive analysis are reported using bar charts to show percentages and the mean values of software requirements engineering. On the other hand, The Spearman's rho shows change requests correlations using Sig (1-tailed), correlation coefficients and N factors.

Ali et al. [17] identify the impact of changing requirements on software projects using impact analysis approach. The objective is to reduce cost, errors, and customer dissatisfaction. A case study is conducted to validate the impact analysis approach. Nolan et al. [18] are of the opinion that requirements of a complex system are always uncertain at the beginning. Requirements uncertainty analysis technique (RUAT) is used to address the problem of uncertain requirements. RAUT technique uses uncertainty identification and mitigating and risk mitigation tradeoff analysis.

Ghosh et al. [19] discuss instability and dramatic changes in requirements even persist after formal approvals or agreements. The instable requirements will negatively impact the cost, schedule and efforts to complete projects. A decision support tool is proposed to estimate the impact of changes in requirements using data of 33 samples. The limitations of reviewed paper are shown in Table 1.

Table 1. The limitations of Related Work.

Title	Limitation
Risk analysis and mitigation steps in different phases [1].	Not all risk factors are listed whereas companies face other risks during system development life cycle (SDLC)
Software risk management: importance and practices [4].	The proposed model is rules of skimming on action to reduce risks while it is not applied on real SDLC phases.
Risk factors in SD phases [5].	Risk factors listed without providing any strategy to face it.
Risk management assessment using SERIM method [6].	The method applied is not focusing on requirements and complexity issues during SDLC process which can affect any phases of SDLC.
Managing risks in the system analysis and requirements definition phase [7].	Strategies addressed for risk factors that are affecting the project failure during analysis and requirements phase only.
Software risk management and mitigation model [8].	It is not shown that how to apply insurance to software and present a solution to manage software risks as mentioned in their introduction.
Software project risk assessment Fuzzy-ExCOM [9].	the accuracy still moderates so improve the accuracy and sensitivity of proposed model are needed
COCOMO estimates using neural networks [10].	The accuracy of neural network is moderates and less than cost estimation as a hybrid approach
Generic model of software cost estimation on; a Hybrid Approach [11]	The accuracy of hybrid approach still moderates and less than if we are adding KPCA based model which will improve the accuracy better.
Improving accuracy in Software Effort Estimation [12].	The result of cost estimation was not covered to compare it with proposed solution that makes some weakness.
Multi stage software project effort estimation [13].	Did not propose any estimation techniques in order to re-estimate during SDLC which is proposed.
Novel effort estimate-ion approach for requirement changes during SD phases [14]	The accuracy of proposed model is moderate and used only for effort estimation.
Tools & methods for software effort estimation using use case points model [15]	Difficult to use UCP values through companies because there are no use case standard.

### 3. Problem Statement

Changes in requirements affect the schedule and cost of software projects. It requires re-estimating of the budget, schedule, effort and resources to reduce risks [10]. The project manager is responsible for taking amicable actions to mitigate the risks. The existing cost estimation techniques do not calculate the risks of changing requirements. There is a need for cost estimation technique to cater the risks those are associated to change in requirements. This paper is written to address the problem taken up in this research.

### 4. The Proposed Solution

The project managers have to re-estimate the effort, time, and cost during the development of software projects due to changes in requirements [16]. The proposed solution is based on three steps to address the

problem taken up in this paper in order to mitigate the risks those are associated to change in requirements.

#### 4.1. Step one-Impact analysis

Analyze the changes in requirement those are ordered from customers to identify the differences in the use case and class models. We propose set of metrics for analyzing the impact of changed requirement in software project based on mathematical analysis.

#### 4.2. Step two-Re-estimate the software project

Re-estimate with traditional estimation techniques and compare it with initially estimated effort, time and budget. We propose metric to compare the results with the initial estimation of a software project.

#### 4.3. Step three-Manage and mitigate the differences

Managing and mitigating the differences in a project estimated cost due to changes in customer requirements. We propose a new model that will help the project manager to manage the project properly and take the right action when the customer needs to change requirements during development.

#### 4.4. Managing and mitigating changing risk

During software development processes the customer may demand new requirement to be added to the software. The changing or adding requirements definitely will affect the development processes and considered as a risk, but that effect may be either positive or negative risk. The manager of the software project should analysis the impact of that changing ordered from customer to identify it's important and knows its effectness to the software project. After that estimate, the cost of new requirement should be taken to know the effectness whether positive or negative effect. Finally, the software project manager estimates the impact and compares it with initial cost estimation of the project. In addition, the action should be taken by the project manager. The propose below Model describes and will help the manger to manage well and mitigate risks of changing in requirement during system development life cycle as well as take an action on that changes from the customer.

- The proposed processes of impact analysis are:
  - Counts the use cases those will be affected;
  - Counts the classes those will be affected.
- The project managers have to identify changing requirement risks in their project. The proposed metrics will measure the effects of software project during SDLC by comparing the number of original use cases and classes with the new use case and classes added or deleted from the system out of changing requested from clients side. The metrics are shown as follows:

1. Number of Use Case Metric (NU) is:

$$\text{NUMetric} = \text{NAU} + \text{NNU} - \text{DeletedU.C} \quad (1)$$

- A number of actual use case (NAU) is the total number of original use case based on the original requirement that presents in the project.
- A number of new use case (NNU) is the number of use case added to the project.

2. Number of classes Metric (NC) is total number of classes after added or deleted new use case which based on customer demand and it can be measured

$$NCMetric = NAC + NNC - DeletedC \quad (2)$$

A number of actual class (NAC) is the total number of original classes based on the original use case that present in the project.

- A number of a new class (NNC) is the number of classes added to the project.
3. After applied above metrics and based on function point analysis method we will get the following equation:

$$NUMetric = Fu, \quad NCMetric = Fc \quad (3)$$

4. Function point method is applied to measure complexity and size by using equations 4 and 5, While,  $F_u$  and  $F_c$  are measured, FP value is measured again if requirements are changed.

$$FP = F_u \times W_u + F_c \times W_c \quad (4)$$

5. A function point is estimated by counting the weights of use cases ( $W_u$ ) and classes ( $W_c$ ) with adjustments factors [2]. The actual weight is always ranging from 4 to 7 for use cases and 7 to 15 for classes. We can adjust the FP value after requirements are changed as shown in equation no. (5).

$$FP_{adjusted} = FP \times (0.65 + 0.01 + \sum Fi) \quad (5)$$

Where; FP: function point

$\sum Fi$ : sum of fourteen factors for each value range (0-5)

6. FP is estimating the size and complexity of software project. It is used to estimate a line of code (LOC) using Backfiring method to calculate new cost estimation (NCE).
  7. The idea of proposed metrics is based on adjusting the results of new estimation in change requirement (NCE) of software project before comparing it with the initial cost estimation (ICE).
  8. It helps to identify the cost estimation through SDLC. It will also to re-count the effort in person month and duration in a month by decrement the time and person from the task that already finished by project team then re-estimate the result to get the adjusted cost estimation. The proposed metrics are summarized as follows.
- New Cost Estimation (NCE) is the total cost of the project after requirements are changed.
    - New Effort Estimation (NEE)
    - New Duration Estimation (NDE)
  - New Cost Estimation Adjusted (NCEA)
  - New Effort Estimation Finished (NEEF)
  - Number of Effort Estimation Adjusted (NEEA)

$$NEEA = NEE - NEEF \tag{6}$$

- New duration already Finished (NDEF)
- Number of Duration Estimation Adjusted (NDEA)

$$NDEA = NDE - NDEF \tag{7}$$

- The proposed model is shown in fig. 3. It allows the project managers to manage and mitigate risks.

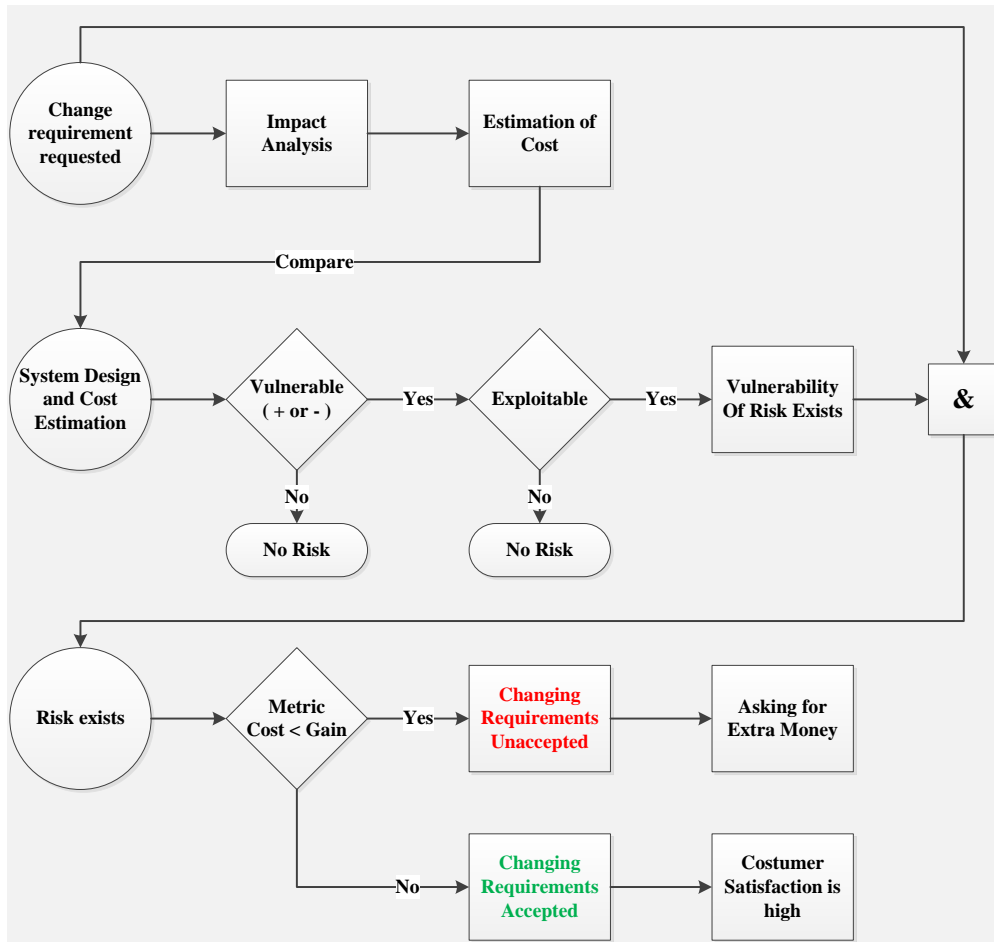


Fig.3. Change Requirement Action Point Model.

- The decision can be taken by using the following proposed metric to compare between NCEA and ICE to get the total effect of the cost, time and effort.

$$\text{Compare (+ or -)} = NCEA \langle \rangle ICE \tag{8}$$



- The project manager will decide to accept or ask for extra money in order to apply the changed requirements using the equation (8).

## 5. Validation of the Proposed Solution

The questionnaire of specific four goals is used in order to validate the proposed solutions which were taken up in this paper. The questionnaire is close ended consisting of 22 questions and the data gathered were based on likert scale method. In addition this questionnaire depends on predefined goals were statistically analyzed to support our proposal. We have collected the data and find the results as follows:

### 5.1. Cumulative statistical analysis of Goal 1.

The first goal is finding and identifying the differences of change requirement risk. Once the project manager identified the differences, it will get help to manage and mitigate the risk. The result of the first goal is shown in Table 2.

Table 2. Cumulative Analysis of Goal 1.

Q. No.	Strong. disagree	Disagreed	Neutral	Agree	Strongly agree
1	1	0	7	18	10
2	2	1	9	20	4
3	0	2	14	18	2
4	0	4	13	16	3
5	0	3	17	14	2
Total	3	10	60	86	21
Avg.	1.67%	5.56%	33.33%	47.78%	11.67%

Table 2 shows the cumulative analysis of identifying and finding the differences due to changing in software requirements. It is shown in Table 2 that 47.78% of the programmers are agreed to this goal, and 11.67% are strongly agreed. Moreover, 5.56% of the professionals disagree and 1.67% of the participants strongly disagree while 33.33% of software engineers remain neutral. Fig. 4 displays the scale for these data.

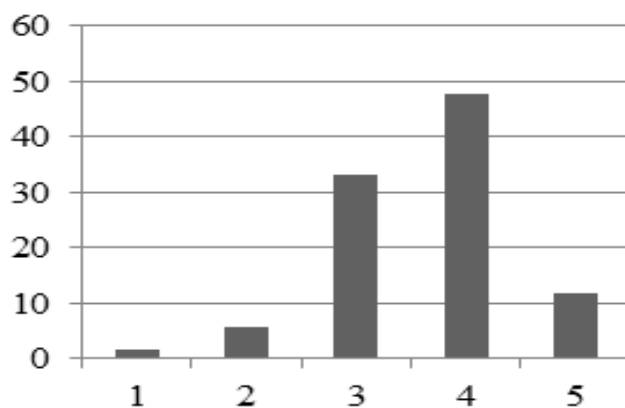


Fig.4. Cumulative Analysis of Goal 1

5.2. Cumulative statistical analysis of Goal 2.

The second goal is re-estimate the changes requirement by using proposed metrics and traditional method. Once the project manager re-estimates the project, it will get help in order to mitigate the risk as well as manage it properly. The results of the second goal are shown in Table 3.

Table 3. Cumulative Analysis for Goal 2.

Q. No	Strong. disagree	Disagreed	Neutral	Agree	Strongly agree
1	1	3	12	15	5
2	1	4	13	15	3
3	1	4	15	10	6
4	1	2	14	16	3
5	1	2	11	17	5
6	2	4	10	15	5
Total	7	19	75	88	27
Avg.	3.24%	8.80%	34.72%	40.74%	12.50%

Table 3 shows that 40.74% of the participants are agreed on this goal, and 12.50% of the professionals are strongly agreed. Moreover, 8.80% of the respondents disagree and 3.24% strongly disagree while 34.72% of the professionals remain neutral. Fig. 5 displays the results of Table 3.

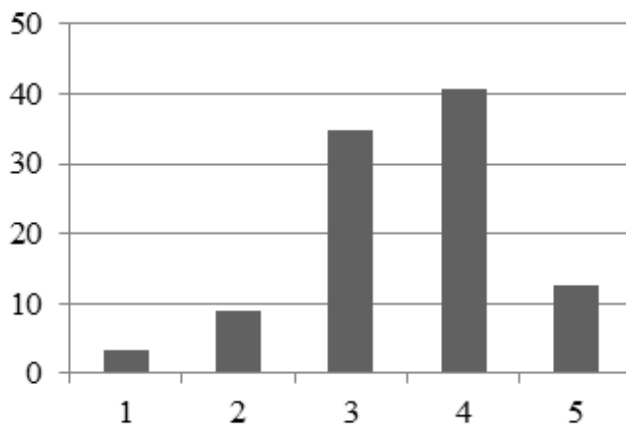


Fig.5. Cumulative Analysis of Goal 2.

5.3. Cumulative statistical analysis of Goal 3.

The third goal is managing and mitigating the risk by using proposed CRAP model. The results of Goal 3 are shown in Table 4.

Table 4. Cumulative Analysis of the Third Goal.

Q. No	Strong. disagree	Disagreed	Neutral	Agree	Strongly agree
1	2	4	6	21	3
2	1	6	7	19	3
3	0	2	11	21	2
4	1	1	9	19	6
5	0	1	20	12	3
6	2	2	16	15	3
Total	6	16	69	107	20
Avg.	2.75%	7.34%	31.65%	49.08%	9.17%

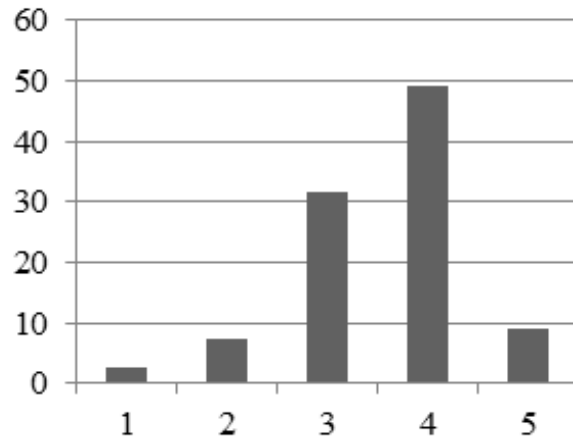


Fig.6. Cumulative Analysis of Goal 3.

Table 4 shows that 49.08% of the participants are agreed on this goal, and 9.17% the professionals are strongly agreed. Moreover, 7.34% of the participants disagree, and 2.75% of the software engineers strongly disagree while 31.65% of the participants remain neutral. Fig. 6 presents the results of Table 4 graphically.

#### 5.4. Cumulative statistical analysis of Goal 4.

The fourth goal is set to deliver software product timely as estimated. The results of Goal 4 are shown in Table 5.

Table 5. Cumulative Analysis of Goal 4.

Q. No	Strong. disagree	Disagreed	Neutral	Agree	Strongly agree
1	1	4	6	20	5
2	1	2	11	16	6
3	1	1	7	18	9
4	1	1	13	16	5
5	0	3	12	18	3
Total	4	11	49	88	28
Avg.	2.22%	6.11%	27.22%	48.89%	15.56%

Table 5 shows that 48.89% of the participants are agreed to goal 4 and 15.56% of the participants are strongly agreed. 6.11% of the participants disagree, and 2.22% of the professionals strongly disagree. 27.22% of the participants remain neutral. Fig. 7 shows this graphically as follows.

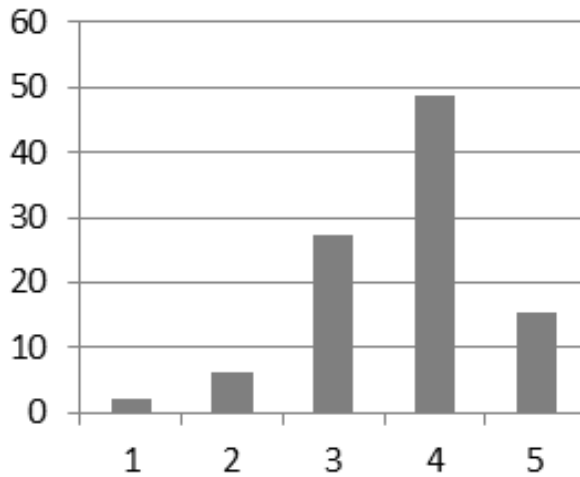


Fig.7. Cumulative Analysis of Goal 4.

5.5. Final Cumulative statistical analysis of 4 Goals.

Table 6 shows that only 2.47% of the software engineers report the very low effect of the proposed solution. 6.95% of the professionals respond the low effect. 31.73% of the respondents report the nominal effect of the proposed solution. 46.62% of the professionals are highly favouring the proposed solution. Among the software professionals, 12.25% of the participants are very highly favouring the proposed solution. As such 58.84% support is available. Fig. 8 displays the results of Table 6.

Table 6. Final Cumulative Analysis of 4 Goals

Likert Scale Goals in %	1	2	3	4	5
Goal 1	1.67	5.56	33.33	47.78	11.67
Goal 2	3.24	8.80	34.72	40.74	12.50
Goal 3	2.75	7.34	31.65	49.08	9.17
Goal 4	2.22	6.11	27.22	48.89	15.56
Total	9.88	27.81	126.92	186.49	48.90
Avg.	2.47%	6.95%	31.73%	46.62%	12.22%

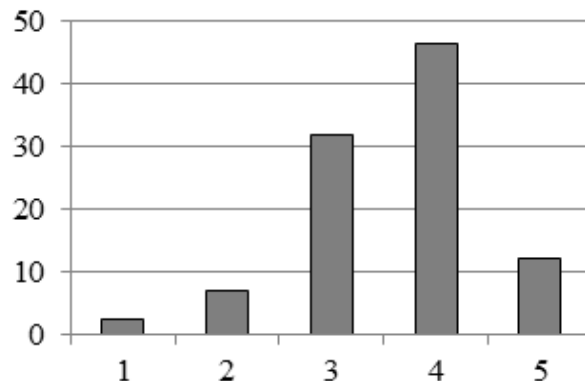


Fig.8. Final Cumulative Analysis of 4 Goals.

## 6. Conclusion

Changing requirements are high risk in software development which may occur by asking for extra or changing in software requirements during its development. Project managers must manage the project and don't leave it vulnerable to any risk factors which can cause project completion failure as estimated and scheduled. Therefore new model proposed in this paper in order to manage and mitigate that risk by using traditional FPA and COCOMO estimation techniques. Moreover, questionnaire for specific four goals was used in order to validate the proposed solutions and how it will aid projects managers to deal properly with changes requirements risks during software development. The result shows that most of the questionnaire participants were agreed on our proposed model. The questionnaire was based on four predefined goals whereas Figure 4 shows the participants agreed on the first goal 47.78%, 40.74% for the second goal in Figure 5, for the third goal were 49.08% in Figure 6 and 48.89% for the fourth goal in figure 7. On other hand disagreed and totally disagreed participants were vary from 1.67% to 8.80% for all goals while neutral participants varied from 27.22% to 37.72%. The questionnaire is designed with 22 close ended questions. The proposed solution will help project managers to identify and finding differences, re-estimate the project, manage as well mitigate risks due to changing in software requirements. Thus deliver the software project to the customers properly as estimated and scheduled while other estimation technique can be used in order to improve the accuracy.

## References

- [1] S. Mittal, "Risk Analysis and Mitigation Steps in Different Phases of Software Development", International journal of scientific research, Volume: 2 |Issue: 6| June 2013 • ISSN No 2277 – 8179.
- [2] Walton Hall and Milton Keynes, *M363 software engineering with object*, ISBN 9780749216191, Open University UK 2008.
- [3] P.K. Suri, P. Ranjan, "Comparative Analysis of Software Effort Estimation Techniques", International Journal of Computer Applications (0975 – 8887), Volume 48– No.21, June 2012.
- [4] A. Al Murad Chowdhury and S. Arefeen, "Software risk Management: Importance and Practices". IJCIT, 2011, ISSN 2078-5828 (PRINT), ISSN 2218-5224 (ONLINE), VOLUME 02, ISSUE 01, MANUSCRIPT CODE: 110740.
- [5] H. Hijazi, S. Alqrainy, H. Muaidi, T. Khmour, "Risk factors in software development phases", European Scientific Journal January 2014 edition vol.10, No.3 ISSN: 1857 – 7881 (Print) e - ISSN 1857- 7431.
- [6] Falahah, "Risk management assessment using SERIM method", International Conference on e-

- Education, Entertainment and e-Management, 978-1-4577-13 82-8/11, 2011 IEEE.
- [7] S. Alqrainy, H. Hijazi, "Managing Risks in the System Analysis and Requirements Definition Phase", International Journal of Computer Applications (0975 – 8887) Volume 99– No.3, August 2014.
- [8] N. K. Rout, N. Parida, S. Mishra, "Software Risk Management and Mitigation Model", JAIEM, Volume 2, Issue 7, July 2013, ISSN 2319 – 4847.
- [9] E. Manalif, L. F. Capretz, A. B. Nassif, "Fuzzy-ExCOM Software Project Risk Assessment" 2012 11th International Conference on Machine Learning and Applications 978-0-7695-4913-2/12 © 2012 IEEE DOI 10.1109/ICMLA.2012.193.
- [10] A. Kaushik, A. Chauhan, D. Mittal, S. Gupta, "COCOMO Estimates Using Neural Networks". MECS I.J. Intelligent Systems and Applications, 2012, 9, 22-28 DOI: 10.5815/ijisa.2012.09.03
- [11] L. V. Patil, Lalit, S. D. Joshi, V. Khanna, "Generic model of software cost estimation on; a Hybrid Approach", 978-1-4799-2572-8/14/\$31.00\_c 2014 IEEE.
- [12] Z. Dan, "Improving the Accuracy in Software Effort Estimation", 978-1-4799-0530-0/13, 2013, IEEE.
- [13] C. Nagar , Dr. A. Dixit, "Multi stage software -project effort estimation", International Journal of Advanced Trends in Computer Science and Engineering, Volume 3, No.1, January – February 2014, ISSN 2278-3091.
- [14] S. Basri, N. Kama and R. Ibrahim, "Novel Effort Estimation Approach for Requirement Changes during Software Development Phase", International Journal of Software Engineering and Its Applications Vol. 9, No. 1 (2015), pp. 237-252, ISSN: 1738-9984 IJSEIA Copyright 2015 SERSC.
- [15] M. Saroha, S. Sahu. "Tools & Methods for Software Effort Estimation Using Use Case Points Model – A Review", ISBN: 978-1-4799-8890-7/15, 2015 IEEE.
- [16] B. W. Muhammad, H. Farah, E. Nadeem, I. Azam, A. Sohail and S. Z. Sheikh, "An Investigation of Changing Requirements with respect to Development Phases of a Software Project," in Proc. Int. Conf. Computer Information Systems and Industrial Management Applications, Poland, 2010, pp. 323-327.
- [17] H. O. Ali, M. Z. Abd Rozan, A. M. Sharif, "Identifying Challenges of Change Impact Analysis for Software Projects", in Proc. Int. Conf. Innovation Management and Technology Research, Malaysia, 2012, pp. 407-411.
- [18] A. J. Nolan, S. Abrahao, P. C. Clements and A. Pickard, "Requirements Uncertainty in a Software Product Line", in Proc. Int. Conf. Software Product Line Conference (SPLC), Munich, 2011, pp. 223-231.
- [19] S. Ghosh, S. Ramaswamy, R. P. Jetley, "Towards Requirements Change Decision Support", 20th Asia-Pacific Software Engineering Conference, 1530-1362/13, 2013 IEEE.

### Authors' Profiles



**Mohammad D. Aljohani** is currently doing master in IT from the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. Aljohani obtained his B. Sc Degree in Information Technology and computing from Arab Open University, at Saudi Arabia in the year of 2011. His interest areas are software requirements engineering, agile software engineering, and CBD.



**Dr. M. Rizwan Jameel Qureshi** received his Ph.D. degree from National College of Business Administration & Economics, Pakistan 2009. He is currently working as an Associate Professor in the Department of IT, King Abdulaziz University, Jeddah, Saudi Arabia. This author is the best researcher awardees from the Department of Information Technology, King Abdulaziz University in 2013 and 2016. He is also honoured as the best researcher from the Department of Computer Science, COMSATS Institute of Information Technology, Pakistan in 2008.

**How to cite this paper:** Mohammad D. Aljohani, M. Rizwan J. Qureshi, "Management of Changes in Software Requirements during Development Phases", *International Journal of Education and Management Engineering(IJEME)*, Vol.6, No.6, pp.12-26, 2016.DOI: 10.5815/ijeme.2016.06.02