

# Non-Technical Issues in Software Designing Phase

Mohammad Ansar

Department of Software Engineering  
Bahria University, Pakistan.  
fa14rpm508@vcomsats.edu.pk

Tamim Ahmed Khan

Department of Software Engineering  
Bahria University, Pakistan.  
tamim@bui.edu.pk

**Abstract**—Software designing phase is an important phase in the software development lifecycle. There could be a number of technical and non-technical issues e.g., communication issues and understand-ability issues which may hinder desired progress in this phase affecting the software project output.

We study how these issues affect software project by conducting interviews and surveys within software houses and IT concerns. We report how communication issues and understand-ability issues contribute statistically significantly towards failure in software projects within organizations.

**Index Terms**—software development life cycle (SDLC), communication issues, understand-ability issues, political issues, software designing phase issues.

## I. INTRODUCTION

Software designing phase in software development is an important phase. User requirements needed to be transformed into software interface designing. Software designing phase has associated problems and issues which can be categorized into technical and non-technical ones. Non-technical problems in designing phase not only affect the software project at later stages but contribute towards project failure if these issues are not managed properly. Non-technical issues during software development are of a wide variety and an assessment to see how they affect overall progress during various phases in software development. Different researchers have investigated the impact of social aspects in developer's work. Social aspects, being an example of non-technical issues software houses are faced in everyday life, impact the developer productivity [28] and affect the software quality [29]. Even project development environment significantly impacts on development team member individually that makes it difficult for team member to achieve socio-technical balance [30]. Poor planning and poor estimation [1] [2] also contribute significantly towards project failure and both induce issues of non-technical nature. Such factors indirectly [3]. Several researchers have evaluated various software project failure criteria. However, there is a requirement to investigate software failure criteria in a phase-wise manner. This would help finding out how software projects get affected by a host of issues at various stages (phases) of software development. Similarly, there is a requirement to establish a relationship between software project failure and non-technical issues. The objective of this research is to see what are non-technical issues in software design phase and how do they impact software development. This paper is organized in the following manner. Background and related work are discussed in Section-II and the problem

statement is explained in Section-III. Research methodology is discussed in Section-IV whereas the findings and analysis of results are shared in Section-V. Finally, the conclusion and outlook is discussed in Section-VI.

## II. BACKGROUND AND RELATED WORK

Software development in a successful manner requires efforts both from software development team and project management managing software development processes. Software project usually starts with requirement gathering and finishes in testing and deployment. In each phase of software development, software development team faces technical and non-technical issues that affect software project and ultimately lead to software project failure. Software organization has an aim to meet the project goal. But due to technical and non-technical issues, it is impossible to meet the software project goal. Software project goals are to develop the software project under specified cost, time and to meet project quality goal. According to [1] [2], software project is affected by poor planning, poor quality control, requirement changing at great variation and poor estimating. In another study [3] authors explain that failure to deliver required and useful functionality, product or process quality, over budget, software project manager loss of interest in software project then there are high chances of software project failure. Communication at all levels is required in software planning phase. It would result in excessive requirement creeping, project scope changing without effective communication between stakeholders. It is impossible to meet project success criteria without incorporating the individual project stakeholder opinion into account [4] [5]. In software development process different studies [6] [7] [8] [9] [10] and [11] emphasize the involvement of user in each phase of software development and take feedback from them. With involvement of user, at one side later stages of project get affected by changing requirement and project scope while on the other side undertaken project leads to failure. Involvement of user in each phase of development enhances project path toward project success and reduces risk of failure. The main reason behind not involving user in development processes are software organization will not always have sufficient resources in project implementation to meet stakeholder requirement [12]. The success of software development highly depends on requirement phase in which project scope is analyzed effectively and efficiently. In requirement phase, if non-technical issues are not managed properly then these issues affect later

stages of software development and ultimately lead to software project failure. If non-technical issues in requirement phase are not managed properly then next phase e.g., software designing and later phases get affected. Such issues in requirement phase are communication issues, understand-ability issues and political issues. These issues get increased in later stages of software development. In software projects, communication receives little attention [13]. According to authors in [14], it is proved that can be significantly improved by using effective communication style and pattern software project productivity. While it is emphasized in [15] that there is a need for adequate communication for user/designer in software project. In case software development team has low understanding of requirements with misinterpretations and in-validated assumptions under which software application is developed, the chances of project success is undermined [16]. If requirement not agreed or confirmed by development team then it results to put worst effect on software project in terms of reduced functionality [17] [18] and lower quality goals [1] [2] [6] [19]. When communication and understand-ability issues are not managed properly as required, the cost of changing project goal becomes significant. This eventually requires project reviews again and again and cost of rework ultimately leads to project failure [20]. These cost overruns [6] [21] [22] contribute significantly towards project failure and deadlines are often missed. In addition to this, when there is no coordination and consultation between business analyst/requirement engineer, the impact is felt on the development team. In case a software project has different stakeholders with different interaction value, attitude, behavioral norm and different communication approaches, it results in miss-understanding and miss-interpretation of project objective and result in conflict, miss-interest and overall impact the project success [23]. The main reason behind is effective communication gap among stakeholder in the software development process. According to [23] [24], for effective collaboration and coordination communication is one of the most efficient method. In software development process, cooperation amongst stakeholders has links with software quality. Authors in [25] state that quality and progress of whole project development is largely affected by efficiency of stockholders cooperation. In another study conducted by [26], challenges to software development organization are identified. It is further explained how these challenges such as lack of communication, lack of understand-ability in team, schedule problem, lack of coordination and lack of sharing knowledge and transfer among teams affect software development. In addition to this, lack of communication channel [27] in software development team make it more difficult to share information and provision of instant support. Another success criteria of successful development of software requires a deep knowledge of corresponding domain [31], when software requirement analyst does not have corresponding domain understanding fully then it resulted in misunderstanding and inappropriate software. Moreover, author in [32] prove that success of software project in terms of completion of software project within time and under-

budget, delivering with better quality and knowledge transfer and it is only possible when right project plan developed [33]. According to authors [34], software project management faces problem of exceeding budget than planned budget and lagging behind schedule that overall resulting in low satisfaction of customers because having project management issues. Author also identified certain issues faced by all software development organizations are requirement misconception, ambiguous and unclear requirements, communication barrier, client side expectation are very high, culture and language issues specially in outsourcing project, development environment and testing issues in distributed environment.. Communication management and understand-ability are two major focused area by project manager and software development team members to get project completion successfully [34]. Author also state that requirement management is weak area in software development life cycle that has been identified by the different researcher. Author [35] found that project success is linked to issues surrounding team assignment on project and specific guidance that are required and provided.

### III. PROBLEM STATEMENT AND RESEARCH METHODOLOGY

We investigate non-technical issues in software designing phase that will affect later stages of software development and/or increases chances in software failure. We divide our work in parts such that we carry out exploratory research (pilot study), in the first part, in which we explore and investigate non-technical issues with the help of nineteen professionals. These professional have experience in managing and implementing software project in their respective companies. In this part of our study, we use snowball sampling technique. The purpose of doing exploratory research is to know more about non-technical issues in software designing phase. After exploratory research, we proposed two hypothesis:

- H1 There is a significant relationship between software project failure and non-technical issues (communication issues, understand-ability issues and political issues) in software designing phase
- H2 Non-technical issues (communication issues, understand-ability issues and political issues) contribute significantly towards software project failure?

We then develop a questionnaire in which we ask following questions and conduct a survey, in the second part of this research:

- Q1 What are non-technical issues existing in your software organization in software designing phase?
- Q2 What are the non-technical issues, if exist in software designing phase causes software project failure?
- Q3 Do you think, on the basis of your experience, non-technical issues software designing phase becomes causes of software project failure?

Q1 and Q2 have sub-questions in which we use information that is found during pilot study from 19 different interviewers

who had previous experience in software development. While Q3 was on binary scale in which response either Yes or No. We use this questionnaire to conduct a survey. This survey is conducted in 30 different software organizations. We only consider those organizations where at least two or more participants were involved in the survey. We select participants in such a manner that they are either at mid management level or they are involved in software design process. Survey is self-administered and help is only provided to respondents in case they do not understand any part of the questionnaire. During survey, we also gather data pertaining to other non-technical issues that are not listed in questionnaires. We ask respondents questions in the form of a short interview to find out other non-technical issues that are not covered in the questionnaire or they are particular to a unique personal experience. After conducting survey successfully, we analyze responses and list into tables from highest to lowest frequency. We use Pearson correlation for statistical analysis. We also conduct logistic regression to know the effect of non-technical issues in software project failure. Our dependent and independent variables are communication issues, understand-ability issues and political issues while our dependent Variable is software project failure (SPF).

Dependent and independent variables are on binary scale and either respondent responded to variable in Yes or No. We visited 30 different software organizations to know about non-technical issues at software designing phase and distributed questionnaire. The selected organizations include larger organization are those who has more than 100 employees, medium sized organizations having between 50-100 employees and small organization having less than 30 employees.

#### IV. RESEARCH FINDINGS AND ANALYSIS

In this section, we describe our survey and statistical analysis results. Besides the issues listed into Q1 and Q2, we were found many other non-technical issues that having the characteristics of worst effect on software project success criteria. Mostly issues facing by these surveyed organization in software designing phase are listed in Table II and shown in Figure 2 that causes software project failure.

After conducting the survey, response of the Q1 shown in Table I and Figure 1. The responses of the question What are current non-technical issues existing in your software organization? are arranged from highest to lowest frequency of occurrences. In which some issues are related to communication issues, understand-ability issues, political issues and other non-technical issues relating to process used in requirement gathering phase.

We present in Figure 1, the most prevailing issues in software organization after analyzing survey responses. In Table II and in Figure 2, the responses of Question No. 2 are shown. We ask issues related questions to the participants which are mainly related to software design phase and highlight causes related to software project failure. We choose three variables related to communication issues, understand-ability issues and political issues. We are now ready to test our hypothesis.

TABLE I  
NON-TECHNICAL ISSUES

S #	Non-technical issues	f(yes)	f(no)
1	Due to time constraint, designing through proper analysis is not carried out	26	3
2	Detailed analysis are not carried out	25	6
3	Do not have enough time for prototyping	24	6
4	Understand-ability Issues	23	4
5	In designing, flowchart, hierarchical diagram does not drawn	16	14
6	Communication Issues	15	15
7	At this level, customer working environment are not analyzed	14	16
8	Design changing at great variation by changing customer requirement	13	17
9	For alternative evaluation, prototype does not constructed	13	17
10	If additional requirement identified at this level, then these are not documented properly	12	18
11	Final design interfaces does not show to customer along development	11	19
12	All software designing process do not documented properly	11	19
13	Changes in design suggested by customer does not documented properly	10	20
14	Along development, design of final interfaces do not show to customer	9	21
15	Designer design software on the basis of experience rather analyzing customer context	7	23
16	Right tools are not available to design	5	25
17	Political Issues	4	26
18	Designer has not required and enough knowledge about designing procedures	3	27
19	Programmer don't participating in designing	1	29
20	Prototyping issues	1	29

For H1, we can explain that there is a significant positive relationship between software project failure and communication issues ( $r=0.364$ ,  $df=6$ ,  $p<0.001$ ) and understand-ability issues ( $r=0.554$ ,  $df=6$ ,  $p<0.001$ ) in software designing phase in software development as shown in Table III.

While in this hypothesis, political issues does not significantly correlated in software project failure. We concluded that software organizations which has communication and understand-ability issues has highest chances of software project failure.

However, understand-ability issues requires more attention than communication issues because this phase requires great understand-ability about client business processes, requires to know about user who use the software, making no difference among expert and novice user, designer dont understand what user demanded, understanding user environment in which software will operated. Communication also requires to enhance software success criteria. Communication issues in designing phase arise when there is no involvement of user in designing phase, user/client feedback are not on time, no communication among requirement, designing and coding team members.

For testing H2 in which we investigate if non-technical issues (communication issues, understand-ability issues and

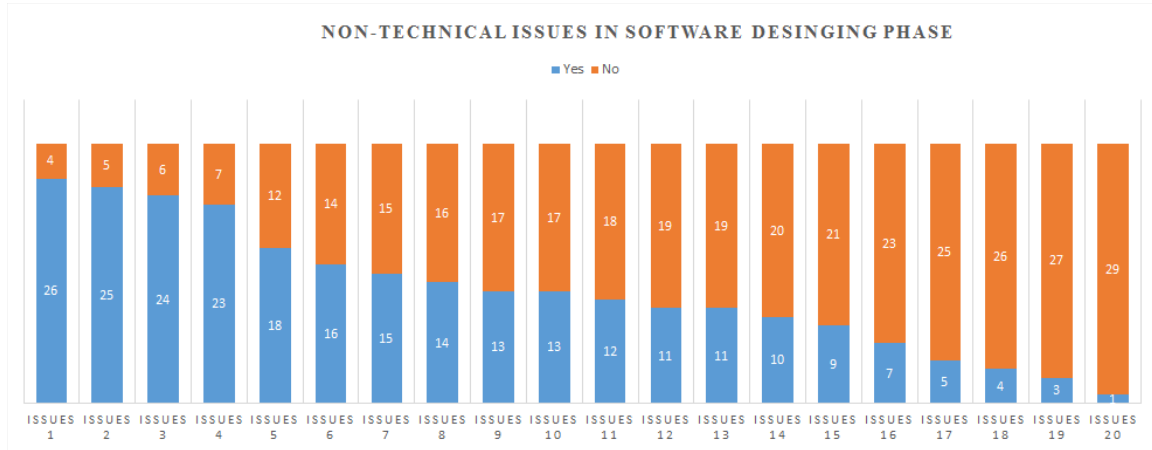


Fig. 1. Design phase issues

TABLE II  
RESPONSES ON NON-TECHNICAL ISSUES

S #	Non-technical issues	f(yes)	f(no)
1	Communication Issues	24	6
2	Software interfaces not designed properly due to time constraint	20	10
3	No communication between software design and requirement teams	18	12
4	Designers design overlooking features	17	13
5	Comment on Design are not on time	15	15
6	Understand-ability Issues	15	15
7	To verify design, customer are not available on time	12	18
8	Design team do not meet user expectation	10	20
9	Use case and sequence diagram does not carried out	10	20
10	Missed information in Designing Tools/Techniques (Use Case, Sequence Diagram, Flow Chart etc.)	9	21
11	Persona is not developed	7	23
12	Different designer working from different team on same assignment	6	24
13	Political issues	5	25
14	Designing tool is not available	4	26
15	Customer demanded more than budget	2	28

TABLE III  
ISSUES SEGMENTATION

	Communication issues	Understand-ability Issues	Political Issues
SPF	0.364	0.554	0.171
Pearson correlation			
Sig(2-tailed)	0.001	0.559	0.143
N	30	30	30

political issues) significantly causes software project failure or not. We apply logistic regression because our dependent variable (software project failure - SPF) has two value Yes and No. Our dependent variable is Question 3 where as our independent variables are communication issues, understand-ability issues and political issues.

Direct logistic regression is performed or testing the H2 to assess the impact of non-technical issues (communication

issues, understand-ability issues, political issues) on the likelihood that the software organization would responded that they had the software project failure on the basis of these issues. The model contains three independent variables (communication issues, understand-ability issues, political issues). The full model containing all predictors was statistical significant, in which chi-square 12.328 (df=3 and N=30) and  $p < 0.006$ . Indicating that model is able to distinguish between software organizations who reported and did not report software project failure on the basis of these non-technical issues. The model as a whole explained between 33.7% (Cox & Snell R Square) and 47.8% (Nagelkerke R Square) of the variance in software project failure and overall correctly classified 80% of cases. As shown in Table IV, only one of the independent variable made a unique statistically significant contribution to the model (understand-ability issues) while other variables communication and political issues did not contributed significantly.

The strongest predictor of reporting software project failure was understand-ability, recording an odd ratio of 16.089. This indicated that those software organizations respondent who reported communication issues were over 16.089 times more likely to report a software project failure than those who did not reported software project failure, controlling for all other factor in the model. Its mean that software project failure, in designing phase in software organizations, causes by understand-ability issues 16.089 times more than other issues like communication and political issues as described by our model. For testing H2 in which we state that does non-technical issues (communication issues, understand-ability issues and political issues) significantly causes software project failure. We apply logistic regression because our dependent variable (software project failure - SPF) has two value Yes and No.

## V. CONCLUSIONS

Designing phase in software development is an important phase of software development life cycle. In this phase, user interfaces, system architecture design etc. are carried out. In

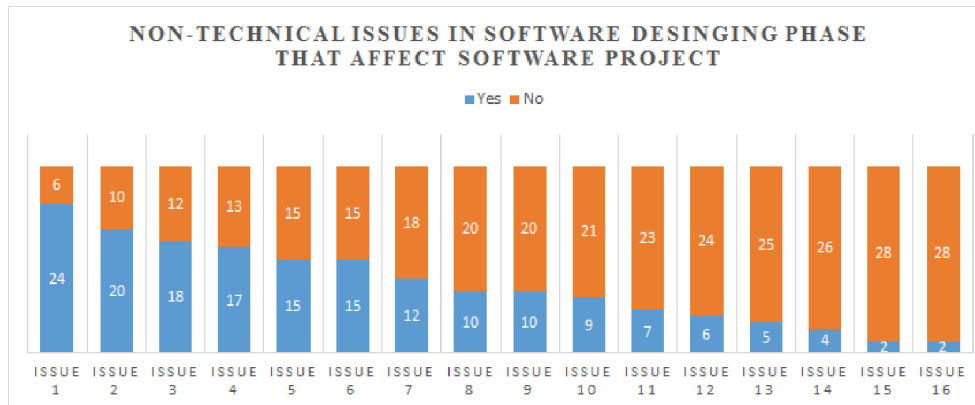


Fig. 2. Design phase issues

TABLE IV  
RESULTS

	B	S.E.	Wald	Df	P	Odds Ratio
Communication issues	0.438	1.391	0.099	1	0.753	1.550
Understandability issues	2.779	1.218	5.209	1	0.022	16.089
Political issues	1.511	1.059	2.036	1	0.154	4.529
Constant	-3.533	1.277	7.653	1	0.006	0.029

this study we have two objectives, first to identify the non-technical issues in software designing process and second to know the effect of communication, understand-ability and political issues on software project failure in this phase. We find out that there is a significant number of non-technical issues. We also find out that improper treatment to non-technical issues leads to project failure while full consideration of such issues is likely to reduce project risks. We surveyed 30 different software organization and found that software development organization facing problem in this phase. Our main purpose behind to know the weather software designing process followed properly. In this phase of software development, these software organizations can be characterized as having communication issues, miss-management in software designing process, no user involvement and not follow properly usability and designing principle and processes (in which usability affected). We also report that issues of highest importance is related to understand-ability and is less tilted towards communication issues that cause software project failure and contribute more adversely on the later stages of software development. However, communication issues also affect software design phase but not as much as understand-ability issues.

#### REFERENCES

- [1] C. Jones, Estimating software costs: bringing realism to estimating, New York: McGraw Hill, 2007.
- [2] C. Jones, Applied software measurement: global analysis of productivity and quality, New York: McGraw Hill, 2008.
- [3] H. Taherdoost and A. Keshavarzsaleh, "Managing Successful IT Project; Marketing Perspective." [Online]. Available: <http://www.asanware.com.my/managing-successful-it-project-marketing-perspective/>. [Accessed 2015].
- [4] N. Agarwal and U. Rathod, "Defining success for software projects: an exploratory revelation," International journal of project management, vol. 24, no. 4, pp. 358-370, 2006.
- [5] J. D. Procaccino and J. M. Verner, "Software project managers and project success: an exploratory study," Journal of system software, vol. 79, no. 1, pp. 1541-1551, 2006.
- [6] R. Schmidt, K. Lyytinen, M. Keil and P. Cule, "Identifying software project risk: an international delphi study," Journal of information management systems, vol. 17, no. 4, pp. 5-36, 2001.
- [7] K. Lyytinen and R. Hirschheim, Information failure a survey and classification of the empirical literature, Oxford university press, 1987, pp. 257-307.
- [8] K. E. Emam and A. Koru, "A replicated survey of IT software project failures," in IEEE Computer, 2008.
- [9] D. L. Hamil, "Your mission, Should you choose to accept it: Project Management Excellence," Geo Community, 2014. [Online]. Available: <http://spatialnews.geocomm.com/features/mesa1/>. [Accessed 2015].
- [10] The Standish Group, "Standish Group survey "CHAOS"," The Standish Group International, 1995-2010.
- [11] I. Attarzadeh and S. H. Ow, "Project Management Practices: The Criteria for Success or Failure," Communications of the IBIMA, vol. 1, 2008.
- [12] J. Soini, "A Survey of Metrics Use in Finnish Software Companies," in International Symposium on Empirical Software Engineering and Measurement, 2011.
- [13] J. J. Elam and D. Walz, "A Study of Conflict in group design activities: Implications for Computer Supported Cooperative work Environments," in in Twenty-First Annual Hawaii International Conference on System Sciences. Decision Support and Knowledge Based Systems Track, Hawaii, 1988.
- [14] Guina, Patricia, Bostrom and R. P, "Development Of Computer-Based Information System: A Communication Framework," DataBase, Spring, 1986.
- [15] T. P. Crona and T. L. Means, "System Development: An Empirical Study of User Communication," ACM SIGMIS Database, Spring, pp. 25-34, 1984.
- [16] E. Bjarnason, K. Wnuk and B. Regnell, "Are you biting off more than you can chew? A case study on causes and effects of overscoping in largescale software engineering," Information and Software Technology, vol. 54, no. 10, pp. 1107-1124, October 2012.
- [17] D. Dalcher, "Falling down is part of growing up; the study of failure in the software," in 7th SEI CSEE Conference on Software Engineering, London, UK, 1999.

- [18] H. Taherdoost and A. Keshavarzsaleh, "Managing Successful IT Project; Marketing Perspective".
- [19] Baccarini and David, "Baccarini D, The logical framework method for defining project success," *Project management journal*, vol. 30, no. 4, pp. 25-32, 1999.
- [20] O. Hitoshi and Z. Lan, "Adapting Agile Methodology to Overcome Social Differences in Project Members," in *Agile Conference*, 2013.
- [21] B. C. Anda, D. I. Sjberg and A. Mockus, "Variability and reproducibility in software engineering: a study of four companies that developed the same system," *IEEE transaction, software engineering*, vol. 35, no. 3, pp. 407-429, 2009.
- [22] Atkinson and R.Sourcer, "Project management: cost, time and quality, two best guesses and a phenomenon, it's time to accept other success criteria," *International Journal of Project Management*, vol. 17, no. 6, pp. 337-342, 1999.
- [23] Shari Lawrence Pfleeger, "Experimental design and analysis in software engineering, Part 1: The language of case studies and formal experiments," in *Annals of Software Engineering*, 1995.
- [24] T. Varkoi, "Development of measurement programs to support process improvement in small software companies," in *Proc. of the European Software Measurement Conference (FESMA99)*, 1999.
- [25] M. Niazi, S. Mahmood, M. Alshayeb, M. R. Riaz1, K. Faisal1 and Narciso, "Challenges of Project Management in Global Software Development: Initial Results," October 7-9 October 2013.
- [26] A. Gopal, J. A. Espinosa, S. Gosain and D. P. Darcy, "Coordination and Performance in Global Software Service Delivery: The Vendors Perspective," in *EEE Transactions on Engineering Management*, 2011.
- [27] H. Huang and E. M. Trauth, "Cultural Influences on Temporal Separation and Coordination in Globally Distributed," in *International Conference on Information Systems (ICIS)*, 2008.
- [28] M. Cataldo, . D. Herbsleb and . M. Carley, "Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity," in *In ESEM 08: Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement*, New York, USA, 2008.
- [29] M. Cataldo, A. Mockus, J. A. Roberts and J. D. Herbsleb, "Software dependencies, work dependencies, and their impact on failures," in *IEEE Transaction on Software Engineering*, 2009.
- [30] M. Zhou and A. Mockus, "Does the Initial Environment Impact the Future of," in *Software Engineering (ICSE), 33rd International Conference*, 2011.
- [31] M. Vigder, "End-User Software Development in a Scientific Organization,," in *SEEUP '09 Proceedings of the 2009 ICSE Workshop on Software Engineering Foundations for End User Programming*, 2009.
- [32] W. U and M. A, "Information systems outsourcing success: a client-service provider gap analysis in Pakistan," *Journal of Information Technology Management*, XV (1-2), pp. 1-13, 2004.
- [33] S. Deshpande and I. Richardson, "Culture in Global Software development - a Weakness or Strength?," *International Conference on Global Software Engineering*, 2010.
- [34] Jalil and Hanif, "Improving management of outsourced software projects in Pakistan," in *Computer Science and Information Technology, ICCSIT 2009 2nd IEEE International Conference* , 2009.
- [35] Root, Rosso-Llopert and Taran, "Key Software Engineering Concepts for Project Success: The Use of "Boot Camp" to Establish Successful Software Projects,," in *Software Engineering Education & Training, 2007. CSEET '07. 20th Conference*, 2007.