IRBAS www.irbas.academyirmbr.com April 2022
International Review of Basic and Applied Sciences Vol. 10 Issue.2

Analysis of Critical Risk Factors Affecting Software Quality A Study of KPK, Pakistan Software Industry

MUHAMMAD IMRAN KHAN KHALIL
University of Engineering & Technology, Peshawar, Pakistan.
Email: imrankhalil@uetpeshawar.edu.pk

AHAD ULLAH
University of Engineering & Technology, Peshawar, Pakistan.

AMER TAJ
University of Engineering & Technology, Peshawar, Pakistan.

IZAZ AHMAD KHAN
Department of Computer Science, Bacha Khan University, Charsadda Pakistan.

FAIZAN ULLAH
Department of Computer Science, Bacha Khan University, Charsadda Pakistan.

FAWAD TAJ
Department of Computer Science, Vrije University, Amsterdam Netherland.

SABIR SHAH
Department of Computer Science, Bacha Khan University, Charsadda Pakistan.

Abstract

The primary objective of this study is to categorize the critical risk factors influencing Pakistan's software development industries, particularly in Khyber Pakhtunkhwa (KPK), including maintainability, team dynamics, software defects, time savings, and staff turnover. All these factors significantly contribute to the software quality of the software industries, which somehow elapsed by the software houses industries in Khyber Pakhtunkhwa. Cronbach's Alpha Reliability (α) is used to test the reliability of variables, accompanied by a regression analysis of the effect of these factors. After investigation, we find out the overall positive and significant association and impact of these factors on adequate software development quality. This research study provided new visions into the software product quality of the software industries, like that a few research has been led in Pakistan to identify these factors that significantly impact the software development quality of the software industries.

Keywords: *Maintainability, Team Relationship, Software Defects, Time Reduction, Employee Turnover.*

Introduction

Software developers and engineers continually look for factors that may impact the software's quality, especially any significant risks. Risk is the likelihood of real hurts, losses, damages, or other undesirable features that could result from internal or external flaws and could be prevented by taking preventive action

for high-quality products (Larsen and Olaisen, 2013). Risk management is identifying, managing, and removing the factors that cause or produce risk-effective problems that can harm our project. It may damage the cost, time, and quality of the project. The five risk variables that could affect software quality are highlighted in this research report. Software product competence and capacity that satisfy all confirmed requirements is called project quality (ISO/IEC 9001) (Khalil et al., 2013). The two fundamental variables—internal and external software quality—can be used to gauge a project's quality. While the exterior software quality is the user expectation and the capabilities when the software is in use, the internal software quality is the quality that is entirely dependent on the code and not apparent to the customer (Tarafdar, 2007). The software houses industries, which engage and provide services to build software or essential goods that are software, play a significant part in managing these requirements. The following software houses come in various types:

- Primary, well-known software producers
- Specialist software houses
- Businesses involved in creating software components like developer express and application servers.

In addition, software firms have a specialized professional team, such as a business recruit, a software developer, and a software tester. This is a must for well-run, capable software businesses (Bock et al., 2005). The obstacles that the KPK software houses encounter here harm the organization's ability to produce high-quality software; therefore, applying this information can resolve the following issue (Hernández et al., 2013).

The impact of employee turnover on software production Employee turnover indicates how long a worker is likely to remain with a company. With a strong sense of cooperation, a project can advance as needed. As technology develops, information technology projects managed by multiple disciplines that collaborate to achieve success face increasingly complex and challenging obstacles (Petter et al., 2012).

The software industries may suffer if there are flaws or consequences that are inappropriate for the computer system user during any computer working environment. It's conceivable for some software bugs to malfunction or to be anticipated errors (Singh et al., 2012). The ability to sustain software dependability is essential for producing high-quality products that can be maintained and cover flaw exclusion.

Software development time reduction requires a controllable management plan. The methodology and procedure for speeding up tasks so they can be completed more quickly while maintaining high standards for the items they produce in response to consumer demand (Kumar, 2012). It would be wise to control the product production process about this critical risk factor for higher software product quality (Mulhearn, 2011).

Research Objectives

1. To prepare a backup plan for the project's product competition.
2. Identify every aspect that can contribute to market drawbacks for comparable competitor items in the same software corporate sector.
3. To determine a different strategy for managing the impact of staff turnover.
4. To determine how the time reduction will affect the software production phase and whether it will negatively influence the software's quality.
5. How to manage software flaws if they are negatively affecting software quality.

Scope of the Study

This study was conducted to identify all the significant risk variables that could impact the software's quality. Based on our project's concerns, we have identified some essential risk factors that include maintenance, collaboration, software flaw, and time-saving measures for employee departures. Thus, this study will assist Peshawar's software companies to improve their strategy for producing high-quality software.

Literature Review

Maintainability

The perceptions that are involved in the maintainability of software quality include the modularity of the software, understanding the software, changes that take place both during and after the development process, testing the software, reusing the software in different contexts, and moving the software from one development team to another development team. Given that when the code was being written, these components did not pose any significant issues (Volberda et al., 2012).

However, the maintainability is poor or not very impressive. In that case, there will be a lot of minor destruction with better coding practice, a strategy for avoiding complexion, and better software documentation practice. This will make a difference between reading the code clearly and easily versus chaotically and making it difficult to read the code (Jorgensen et al., 2006). At the software development stage, immediate attention must be paid to software maintenance to condense the budget for software effectively. Throughout the lifespan of software, a few changes may occur for the reasons listed below (Ievtushenko and Hodge, 2012):

- The occurrence of differences in software requirements
- The identification of software as having flaws
- Variations in the environment of the software
- Variations in the hardware needed for the software also assert a variance in the software.

Software Defect

It is a failure of some form caused by software bugs, an anticipated error, or faults in the computer system that may cause flaws and incorrect results for the computer system user while they are using the computer in any computing environment (Hamdan, 2013). The following terms are used to further the expression of terminology for software defects.

Error: This relates specifically to the software development phase when mistakes, misconceptions, and misinterpretations have happened and negatively impacted software quality. The errors during the development phase are the primary cause of the potential lack of total focus on the task (McLeod et al., 2011). Hence the developer categories are as follows: Software engineers, developers, and analysts are all involved in software testing. Therefore, for better planning, coding, and testing to produce the desired results, everyone involved should pay close attention and concentrate (Willcocks, 1991).

Fault: A flaw in software that causes it to create entirely erroneous results, deliver completely incorrect output, produce an error, or prevent the software from performing as it should for what it was designed to do (Dejaeger et al., 2011).

Failure: Failure describes a component or software system that demonstrates a lack of ability to carry out a task in a given working environment (Bannerman 2008). Some of these include control methods for software flaws or indicators that point to the causes of the defects. A unique technique or process

frequently used during the coding phase and whose origins are relatively similar to those of software faults denotes a defect or error not in the system itself but in the source code that produces the failure in the end product. It is not a straightforward or easy task to identify software flaws; instead, it requires a manageable pattern that can be used to regulate the software faults, which are listed below (Ali et al., 2012).

Team Relationship

Maintaining efficiency in software quality is a difficult issue for every software firm nowadays. Software-based products have a significant impact on how elegantly individuals live their lives. With the development of technology, the administration of information technology projects by multiple disciplines that work as a team presents even more challenging difficulties. Teamwork is essential to practically all IT tasks (Wu, 2013). Teamwork is crucial for many software quality assurance and development project departments, so every firm needs to consider it fully. This is also critical for the process of sound on a project. The project manager's and the team's leaders' responsibility is to carry out any of their jobs in a way that leads to the project's quality meeting the specified standards (Imanara et al., 2012). Additionally, pay attention to how each act in a way that impacts team trust. Numerous academics have noted that trust is the key to many different characteristics and that when it is lacking, teamwork suffers, leading to project failure and poor quality. Other team characteristics like leadership, Communication, and self-managing teams also emphasize the value of trust (Khan et al., 2013). In simulated teams, trust is also mentioned. The current focus for many researchers and specialists is on efforts to identify team success and various team issues, such as Communication, management, and trust (Alteminu et al., 2012). A major issue that determines the success or failure of any endeavor is teamwork. Control, organizational responsibilities, and people are other factors that contribute to the importance of workgroup cooperation and impact team effectiveness. A wide-ranging study can be started in the domain of project team management and the problems that arise within teams that result in team failure and, in turn, in IT project failure or subpar software quality. The lack of realistic studies on issues relating to people, such as trust, Communication, conflicts, bother, etc., is one of the non-nominal studies that are related to the reasons for the project quality disaster (Reich et al., 2012).

Time Reduction

The cycle time required for one complete process, action, or any work from the beginning until the finish can be stated as the notion of cycle time reduction. The total amount of time that passes from the beginning to the conclusion of an activity or operation is cycle time (Habib and Ashfaq, 2013). The procedure and methodology for time reduction, or to shorten the time required to accomplish a task in a way that increases productivity and improves product quality based on customer demand (Jan et al., 2012). Any company that wished to compete with its products had to cut down on productivity time. To do this, a procedure that helped cut down on the organization's overall product throughput time had to be adopted. Therefore, time compression is crucial to a product's cost and production level.

When the task's time cycle is shortened, productivity will steadily rise. A company can generate more valuable goods with fewer resources if productivity increases, boosting organizational growth (Heitlager et al., 2007). In the past, the software industries focused entirely on the software development cycle cost of product development. Still, today it is crucial for all software companies to pay full attention to this factor.

Employees Turnover

Employee turnover indicates how long a worker is likely to remain with a company. The employee turnover for the distinct company and for their industries stately as a whole high turnover of any organization or company as compared to the competitor relative in the same field means that the organization, due to the employee turnover rate, has a shorter average of the tenure as compared to the same industries which working for the qualitative product the high turnover rate of employee also risky for

the company production if the skillful worker or developer leaving the industries its will be the cause of massive impact on the company development productivity (Huberman, 2010). We can also establish internal and external turnover states here (Simpson et al., 2004). Two aspects of turnover are internal and external. Internal turnover is when individuals quit their current positions and take on new responsibilities inside the same firm.

Internal relocation can be summed up as internal turnover. External turnover: we mean that they left the organization for another attractive job relative to the same field, which may make affect the current progress of any running project with the same employee going the giving task, which causes harmful effect on the software development process that is not a good sign for the quality assurance of any products (Tesch et al., 2007).

Conceptual Framework

This research project's primary goal is to make educated guesses about the impact of employee turnover, time reduction, maintainability, software defects, and team dynamics on software development quality in KPK Peshawar, Pakistan. These five elements significantly influence the quality of the software produced by KPK's software houses. Therefore, these five parameters have a direct impact or relationship with the program's quality and serve as the foundation for this study. The five factors are (1) maintainability, (2) team dynamics, (3) software defects, (4) time reduction (5 Employees Turnover). The information gathered for these five elements from the KPK Peshawar, Pakistan, software house industries.

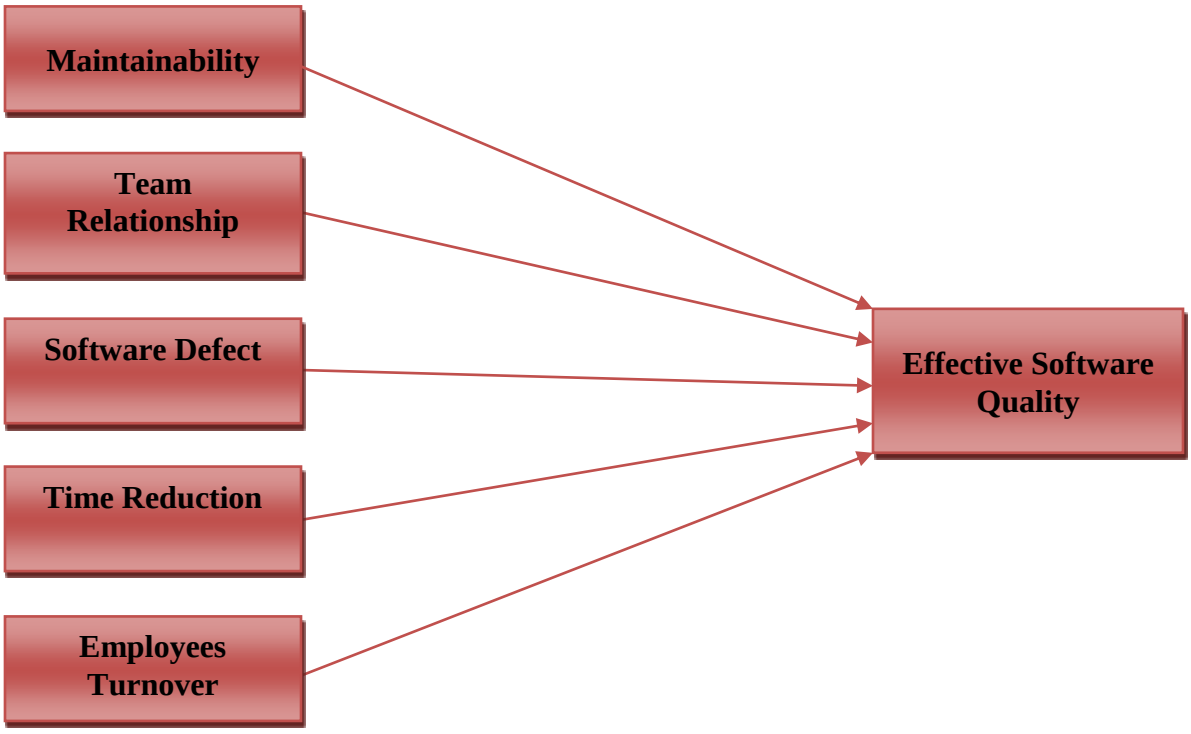

Figure 1: Conceptual framework of the study

Hypothesis

H1: Software quality is negatively and insignificantly impacted by maintainability.
H2: Team relationships negatively and insignificantly impact the program's quality.
H3: Software defects have a detrimental but minor impact on software quality.
H4: Time savings significantly and favorably affect software quality.
H5: Employee turnover has a favorable and considerable impact on the caliber of software.

Proposed Methodology

This study is grounded in quantitative information. A preferred approach to assessable research has been developed to evaluate maintainability, team dynamics, software defects, time savings, and employee retention as they relate to organizations and how they affect the software industry's quality. Additionally, both primary and secondary data have been gathered. The preliminary data obtained through a questionnaire allowed for the collection of data from the top executive software companies of the KPK Peshawar, Pakistan, based on their performance during the previous three to four years. Additionally, secondary data will support literature on maintainability, team dynamics, software defects, and time management. Employee turnover is a problem for organizations and impacts the quality of software companies' software.

Additionally, this research on the foundation of law, such as generalization, will use relativistic philosophy research. It is essential that the respondent fully comprehend the correct meaning of every component that has an immediate bearing on the caliber of the program. Simple random sampling is used in sampling techniques. The sample size is 100.

Data Collection

The cornerstone of the data collection process for this study investigation is the distribution of questionnaires. To collect primary data, questionnaires were distributed to all KPK software enterprises, including junior, medium, and senior-level offices of software houses operating in the Peshawar region of KPK Pakistan. Respondents can quickly grasp the responses to questionnaires because they have been given clearly and understandably. The rickets scale was applied to the questionnaires used in the research study to provide pertinent data.

Population of the study

The number of people from which a sample is taken might be considered the population in this context. The total number of employees or developers employed by the software companies in the Peshawar region is referred to here as the study's population. In the software businesses of the Peshawar region, N: ALL software houses have employees.

Method for Data Analysis

Data analysis is carried out with SPSS version 20. Tables and simple percentages are the methods used to analyze the variables. Regression Analysis is employed in hypothesis testing. Every variable has also undergone a reliability test. To determine the relative significance levels between the independent and dependent variables, linear regression is used for the hypothesis.

Level of Significance

Since there is a 5% level of significance, there is a 95% degree of confidence in the results.

Reliability Analysis

Cronbach's Alpha for the scale of maintainability is 0.716, indicating a high level of internal consistency. In contrast, the reliability of the second scale team relationship is 0.748, meaning internal consistency, and the reliability of the remaining variables, such as software defect, time reduction, and employee turnover, is 0.766, 0.688, and 0.862, respectively.

Table 1: Scales and their reliability

Scale	Cronbach's Alpha Reliability (α)
Maintainability	0.761
Team Relationship	0.748
Software Defect	0.766
Time Reduction	0.688
Employees Turnover	0.862

Analysis, Findings, and Discussions

Analysis

The value of R, which represents the simple correlation between the dependent and all of the independent variables, is shown in Table 2. For example, the correlation between maintainability and software quality is 0.094, which means that they vary together 9.4 percent of the time, just as team relationship, software defect, time reduction, and employee turnover are 30.4 percent, 18.7 percent, 39.4 percent, and 57.6 percent of the time, respectively. The effect of the independent factors on the dependent variable is represented by R^2. For example, the impact of maintainability on software quality is 0.09, which means that variation due to maintainability accounts for 9 percent of the total variation in software quality. In contrast, team relationship, software defect, time reduction, and employee turnover account for 9.2 percent, 3.5 percent, 15.5 percent, and 31.6 percent, respectively. The value of sig. (P), which indicates that at 5%, all variables are statistically significant. Therefore, we accept hypotheses H1 to H5. That maintainability, team relationship, software defect, time reduction, and employee turnover positively and significantly affect adequate software quality at a 5% confidence level.

Table 2: Model Summary

Dependent Variable	Independent Variables	R	R^2	$Sig.(P)$
Effective Software Quality	Maintainability	0.094	0.09	0.003
	Team Relationship	0.304	0.092	.000
	Software Defect	0.187	0.035	0.001
	Time Reduction	0.394	0.155	0.000
	Employees Turnover	0.576	0.331	0.000

Findings

The results of this research work are as follows.

The following factor produced different and unique results for the KPK software businesses (Khyber Pakhtunkhwa). They can make decisions for improving the software house industries based on these. The aspect of software maintainability negatively and insignificantly impacts software quality. For a high-quality product to be maintained and cover the exclusion of flaws, maintainability is essential to preserving software reliability.

On the other hand, the team dynamic has a negligible and minor impact on the caliber of the program. However, it also demonstrates that projects can only advance as needed with a strong sense of cooperation. As technology advances, information technology projects managed by a variety of disciplines that collaborate well to overcome problems make success even more difficult. We can thus adjust this aspect to ensure the project's working continuity using the information from this investigation.

Software quality is negatively and negligibly correlated with software defects. It's similar to a momentary ambiguity in a software program that can manage based on demand. Cutting back on time has a beneficial effect and a significant influence on software quality. However, software development time reduction requires a controllable management plan. The main topics of this study are the methodology and process for speeding up tasks to increase productivity and improve the quality of the products in response to client demand. This research asserts that time reduction is a more critical factor for the software industries of the KPK Peshawar to improve software product quality and offers guidance on managing the product production process to maintain quality.

Employee turnover has a favorable and sizable impact on software quality, which brings us to the final element influencing software quality. The effect of employee turnover on software production Employee turnover indicates how long a worker is likely to remain with a company. We concluded that these two factors significantly impact higher quality and software use life since they are directly related to the productivity of the project development process. It is crucial to managing running projects on time. It was shown that happy employees work better and are more productive in KPK's software house businesses. They enjoy the confidence of their coworkers, who give them the training they need through technological advancements in production.

Discussion

This research effort's central significance and goal were to identify the factors that influence software quality in the Peshawar, KPK, Pakistan region. As a result, we have chosen five vital essential characteristics that have a lot of influence on the software quality of the Peshawar region software houses businesses out of the numerous factors that might affect it both positively and adversely, as well as in terms of its relevance and insignificance.

The factors that we chose are as follows: maintainability, team dynamics, software flaws, time savings, and employee retention. The second key goal of this study project is to ascertain the relationship between software quality in the Peshawar region and maintainability, team dynamics, software flaws, time savings, and employee turnover. Five major hypotheses were resolved during the literature review. We used the following five main hypotheses in the project. The first hypothesis sought to ascertain the influence of maintainability on the quality of the software.

In contrast, the second sought to determine the effect of team dynamics on the quality of the software, the third sought to ascertain the impact of software defects on the quality of the software, and the fourth sought to determine the influence of time reduction on the software quality. The final sought to ascertain the impact of employee turnover on software quality. To collect primary data, one hundred fifty questionnaires were given to male and female junior, medium, and senior level employees of the various software houses industries in KPK Peshawar, Pakistan.

The surveys contained the essential demographic components of Age, Gender, Cadre, Education, Work Type, and Marital Status. The independent variables were Maintenance, Team Relationship, Software Defect, Time Reduction, and Employee Turnover (software quality). The five-point Likert scale was used to grade the questionnaires. Through SPSS, the statistical validity of all five variables was examined. Each variable's results were presented individually. The results of the first three variables were inconsequential, but the other two were highly significant. These five elements' reliability test results revealed that all

variables are trustworthy. The Regression test was then applied to each of the hypotheses, showing the impact of significant and insignificant results on each one separately based on the outcome of the ideas.

Conclusion

We learned how these five factors affect the software quality of the software houses in the Peshawar region in this research project, as well as their primary causes and consequences. After putting the necessary methodologies to manage the impact of software quality effects into practice, we can now control these effects. We have almost reached the desired outcome, which positively impacts the software quality of the software houses in the Peshawar region. We have also provided the necessary methods and techniques for the software houses' employees, developers, and owners to improve the software quality to user demands. The discussed variable or factor—team dynamics, software defects, time savings, and staff turnover—is the basis for our research. We may increase software quality so that it is considerably better for software users than it was in the past if all of these factors are managed effectively and by the needs of the Peshawar region's software users.

All employees in these software businesses are accommodating to these factors, which helps them maintain the high caliber of their software. However, to improve the software quality in the software hoses industries, they should share their knowledge and experiences to foster trust and honesty among their coworkers.

The main goal of this research study is to attain effectiveness towards higher software quality. If regulatory or financial rewards were offered for these many elements, they would produce positive results and performances. These five hypotheses' tests as a whole would be helpful because they can lead to other executions as well.

The software companies' staff turnover rate will go down. The working atmosphere for those who work for software houses will be trustworthy. More individuals with advanced technological knowledge and creative ideas will enter the software houses industries. These software companies will bring in foreign investors and international projects, boosting their productivity.

Recommendations

This research study we conducted revealed several significant findings, and some people are acting on the advice of those findings. To lessen the lousy impact and boost the excellent effect of these elements, these software house businesses should implement new and significantly better technical methods. Software companies should actively foster knowledge sharing among the organization's staff to create a solid and reasonable communication channel. Software companies must have access to career counseling programs to raise employee professionalism and skill levels. Software house enterprises should actively assist employees by creating a welcoming environment where they may work without hesitation. To achieve and complete the desired goal and target and have a positive impact on the software industries, software house industries offer a practical and efficient work plan.

References

Ali, A. A. (2012). Influence of Employee Attitude And Performance Appraisal On Employee Performance. *City University Research Journal.*

Alteminu, M. A. H., Zakariaf, M. S., & Mahmood, A. K. (2012, June). Assessing the performance of information technology strategic planning for organization using performance measurement framework. In *2012 International Conference on Computer & Information Science (ICCIS)* (Vol. 1, pp. 164-169). IEEE.

Bannerman, P. L. (2008). Risk and risk management in software projects: A reassessment. *Journal of systems and software*, *81*(12), 2118-2133.

Bock, G. W., Zmud, R. W., Kim, Y. G., & Lee, J. N. (2005). Behavioral intention formation in knowledge sharing: Examining the roles of extrinsic motivators, social-psychological forces, and organizational climate. *MIS quarterly*, 87-111.

Dejaeger, K., Verbeke, W., Martens, D., & Baesens, B. (2011). Data mining techniques for software effort estimation: a comparative study. *IEEE transactions on software engineering*, *38*(2), 375-397.

Ferguson, C., Green, P., Vaswani, R., & Wu, G. (2013). Determinants of effective information technology governance. *International Journal of Auditing*, *17*(1), 75-99. *L*, *2*(2).

Habib, B., & Ashfaq, R. A. R. (2013). Relationship between factors of quality models and the system development life cycle. *International Journal of Computer Applications*, *81*(10).

Hamdan, K., Belkhouche, B., & Smith, P. (2013). The influence of leadership and work culture on software cost effort. *Journal of Software*, *8*(6), 1353.

Heitlager, I., Kuipers, T., & Visser, J. (2007, September). A practical model for measuring maintainability. In *6th international conference on the quality of information and communications technology (QUATIC 2007)* (pp. 30-39). IEEE.

Hernández-López, A., Colomo-Palacios, R., & García-Crespo, Á. (2013). Software engineering job productivity—a systematic review. *International Journal of Software Engineering and Knowledge Engineering*, *23*(03), 387-406.

Huberman, B. A., & Wilkinson, D. M. (2010). Fluctuating Efforts and Interdependencies in Collaborative Work. *Group Decision and Negotiation*, *19*(2), 169-191.

Ievtushenko, O., & Hodge, G. L. (2012). Review of cost estimation techniques and their strategic importance in the new product development process of textile products. *Research Journal of Textile and Apparel*.

Imanara, Y., Nakajima, D., Samejima, M., Akiyoshi, M., & Morihisa, H. (2012, October). An identification method of risks in project plan document by automatic acquisition of risk expression. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 1235-1239). IEEE.

Jan, M. A. (2012). The Effect of Motivation on The Performance of Field Force In Helicon Pharmaceutical Company. *City University Research Journal*, *2*(2).

Jorgensen, M., & Shepperd, M. (2006). A systematic review of software development cost estimation studies. *IEEE Transactions on software engineering*, *33*(1), 33-53.

Khalil, M. I. K. (2013). Job Satisfaction and Work Morale among Ph. D's: A Study of Public and Private Sector Universities of Peshawar, Pakistan. *International Review of Management and Business Research*, *2*(2), 362.

Khan, A. W., & Khan, S. U. (2013). Critical success factors for offshore software outsourcing contract management from vendors' perspective: An exploratory study using a systematic literature review. *IET software*, *7*(6), 327-338.

Kumar, D. S. (2012). An analytical study of job stress among software professionals in India. *International Journal of Research in Computer Application & Management*, *2*(3), 65-70.

Larsen, T. J., & Olaisen, J. (2013). Innovating strategically in information and knowledge management: Applications of organizational behavior theory. *International Journal of Information Management*, *33*(5), 764-774.

McLeod, L., & MacDonell, S. G. (2011). Factors that affect software systems development project outcomes: A survey of research. *ACM Computing Surveys (CSUR)*, *43*(4), 1-56.

Mulhearn, M. F. (2011). *Tailoring small IT projects in the project planning phase*. George Mason University.

Na, K. S., Li, X., Simpson, J. T., & Kim, K. Y. (2004). Uncertainty profile and software project performance: A cross-national comparison. *Journal of Systems and Software*, *70*(1-2), 155-163.

Petter, S., DeLone, W., & McLean, E. R. (2012). The past, present, and future of "IS success". *Journal of the Association for Information Systems*, *13*(5), 2.

Reich, B. H., Gemino, A., & Sauer, C. (2012). Knowledge management and project-based knowledge in it projects: A model and preliminary empirical results. *International Journal of Project Management*, *30*(6), 663-674.

Singh, P., Suar, D., & Leiter, M. P. (2012). Antecedents, work-related consequences, and buffers of job burnout among Indian software developers. *Journal of Leadership & Organizational Studies*, *19*(1), 83-104.

Tarafdar, M., Tu, Q., Ragu-Nathan, B. S., & Ragu-Nathan, T. S. (2007). The impact of technostress on role stress and productivity. *Journal of management information systems*, *24*(1), 301-328.

Tesch, D., Kloppenborg, T. J., & Frolick, M. N. (2007). IT project risk factors: the project management professionals perspective. *Journal of computer information systems*, *47*(4), 61-69.

Volberda, H. W., Van Der Weerdt, N., Verwaal, E., Stienstra, M., & Verdu, A. J. (2012). Contingency fit, institutional fit, and firm performance: A metafit approach to organization–environment relationships. *Organization Science*, *23*(4), 1040-1054.

Willcocks, L. (1991). Project management for information systems. *Journal of Information Technology*, *6*(3), 238-238.