Automated Risk Analysis Model for Software Development Enhancement

Sundas Akram[1], Muhammad Shafiq[2] and Muhammad Azeem Akbar[3]
[1,2,3]Department of Computer Science, University of Agriculture, Faisalabad, Pakistan
[2]nicepakistan@gmail.com, [3]azeem.akbar@ymail.com

***Abstract*– Risk in Software Programming is the misfortune in programming procedure and instability in the project. Numerous product apparatuses being utilized since long time for execution investigation, testing, check, troubleshooting and constructing applications. Software Engineering process models range from easy to complex like COTS to IDEs. Some of utilized as a part of particular periods of Software Development however others are utilized all through the procedure. Risk Analysis and Management (RAM) is the procedure utilized from starting to end to relieve the risk. Automating risk analysis model with robotized risk tool forecast might be done to moderate risks further. The purpose of this study is to show the analysis and management in Software Development for the flexible and traditional proposing a model for risk analysis, which will reduce risk in automated mode from starting to finish in various stages to much apprentice software engineers in their development.**

***Keywords*– Risk Analysis, Risk Assessment, Risk Management and Software Engineering**

I. INTRODUCTION

Risk has been included in each period of programming advancement process. The effect of risk is exchanged to the next stage. So it results toward the end in the quality bargain, poor administration, and calendar slip, direct impact of the expense of the venture or the notoriety of the association. By and large the risk results issues in the monetary allowance, staff, course of events, prerequisites and due dates. There are civil arguments about the complete meaning of the risk additionally few models offer risk examination yet the vast majority of the models don't offer.

Risk might be explained as the probable loss as overdue task, less benefit, expanded cost, low quality, poor consumer satisfaction and so forth. Risk can be decreased amid the Software Development by tight arranging and assessment. Software risk evaluation is the procedure toward classifying, analyzing, and organizing the risks that may affect the development. Essential steps ought to be taken amid programming improvement to minimize risk.

Essential capacity of programming improvement is to decide the "request of the stages required in software development" and the "foundation of transition criteria" to advance stage to organize. It incorporates criteria for finishing of current stage and decision criteria and the passageway criteria for next stage. The procedure model addresses the product venture questions i.e. what should we do next and to what extent we might keep on doing it [1].

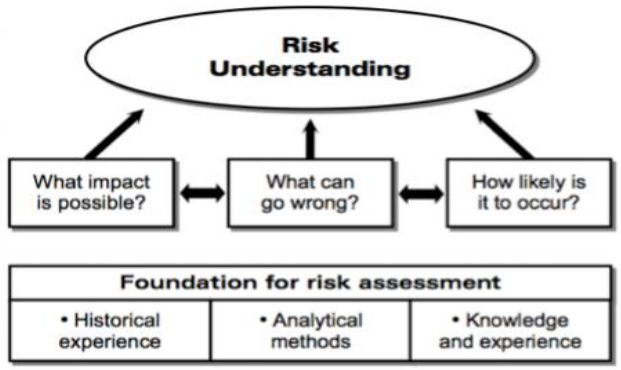

Fig. 1. Risk Understanding and foundation for risk assessment

II. PURPOSE OF THIS STUDY

The existing risk analysis and management tools are precise for software engineering model used for programming. There was a space to develop a tool for independently analysis of the system and software process model tool either active. For efficient and useful risk management, there should be smart and clever risk analysis model which can analyze and implemented to any tool. The purpose of this article is to present the recent review and literature for the need of the said tool and developing an improved model for risk analysis along with its steps.

Therefore, process model contrasts from other programming techniques (methodology) in which its essential center is route through every stage (decide control, information, utilizes; pecking orders; portion necessities; apportioning capacities) and how to speak to stage items (jolt reaction strings; structure outlines; state move diagrams).Why the procedure models are imperative? Since they give direction on the request (increases, stages, approvals undertakings, models and so forth.) in which the venture ought to complete its errands. Numerous activities, as the following segment appears, have ended up badly as they tightened their different advancement and development stages in wrong request.

III. LITERATURE REVIEW

Advancing portable stages and related open doors, various designers have entered the space of versatile application advancement, prompting spaghetti code and different deformities. The paper concentrates on developing region of risk investigation with reference to programming building, gives understanding being developed of uses for cell phones, and manages issues identified with related risks. It likewise depicts the diverse situations in which a portable application works, variables which influence its execution and best practices for versatile application improvement [2].

Risk Identification apparatuses are normally settled for maintaining a strategic distance from or minimizing issues, prone to happen amid programming advancement. It can be expressed as the undertaking of investigating and dealing with the effect of each essential risk happened in the task. With regards to Risk Identification device hones, they built up a checking study, going for breaking down the present situation of Risk Identification hones in programming advancement process. They investigated diverse studies distributed by the most imperative venues distributed up to the year 2013. In light of the examined information set, we outlined an arrangement of helpful strategies and apparatuses for applying Risk Identification in programming ventures. The investigation demonstrates that the majority of the concentrates subjectively depict approaches to assess risks, rather than furnishing peruses with subtle elements on how Risk Identification is to be performed. Such discoveries indicates out the need of further research in the field of Risk Management, particularly for the distinguishing proof of risks in programming improvement process for better results[3]. Risk administration is basic to the accomplishment of any product venture. The venture calendar is the center of the task arranging. In the product venture improvement process, risk booking is a standout amongst the most noteworthy orders that can't be aced by anybody. Along these lines, assessing risks to the calendar is unpredictable. This paper presents distinctive procedures for calendar risk examination [4].

There are distinctive primary purposes of an association administration. Risk administration is one of them. Risk target are constantly productive in all collection. Risk is constantly transported for the achievement of the diverse exercises. In the association it references to the all exercises. To discover a risk and its answer is the principle advantage of risk administration. The association quality can augment by the great administration of the risk. This great administration can likewise enhance the achievement apportion then its disappointment, when the risk is access.

To conquer the assets accessible every one of the exercises of risk lessening required legitimate cost time and calendar. In spite of the fact that another and quick framework is expected to deal with those exercises inside the expense and given time. It is vital for a practical determination. In this paper another system is presented. This method is presented as vital risk diminishment (SRR). This is for ideal risk lessening. Risk is lessen for the wage which is normal by giving the authorization of per step measure to the per risk. Past work is likewise Ykhlef incorporated into it. Here is another strategy which is known as DDP means imperfection location and counteractive action. It is an intense instrument. It is utilized for selecting the diverse risk. These risks have a profound effect and prerequisites. The data which we have in the consequence of this strategy is extremely valuable. This data can likewise be connected in numerous stages. Here and there in the underlying phases of the product venture there are numerous decisions where this data is utilized. So this strategy is extremely helpful in the related ventures. So this application is exceptionally valuable in numerous related ventures [5]. Software Development Life Cycle (SDLC) is helpless against various sorts of risk components. Distinguishing and understanding these risks is a preparatory stage for overseeing risks effectively. The paper introduces a complete hypothetical investigation of the real risk components debilitate each of SDLC stages. A thorough rundown of 100 risk elements was delivered. These rundown mirrors the most oftentimes happening risk figures that are regular to most programming improvement ventures [6].

IV. RISK MANGEMENT

A) Sources of the Risk

Success ratio of an undertaking can be watched that the amount it meets the prerequisite of its client and the amount it is close to its motivation. The accompanying inquiries are for the most part considered in connection to any association.

- Does your software team have awareness about any type of risk?
- Does your team know how to solve or treat these risks?
- Is Project cost, budget estimation even misunderstands?
- Does your team positively response if there is any change in requirements during the project?
- Is it easy to have a clear understanding of customer or stakeholder's requirements?

Fig. 2. Risk Management

B) Risk Management Approaches

From many years, lot of thoughts, models, and practices were printed for the risk analysis and management but

unfortunately, these models were not useable for the software business industry. According to Johnson, usually, there are two types of risk management move towards be able to acknowledged that are accepted for business and risk adjustable. Software business is re-active and used for general problems but second one is pro-active that deal with responsive problems.

C) Risk Assessment

It is the risk acknowledged with the help of evaluating qualitative and quantitative values of the risk also called risk hazard. It's the evaluation of the objectives and the uncertainties that may occur due to risk factors at the end product. It may be very difficult to measure potential loss and uncertainties in project. There may be high chance of error in measuring these two elements. Dealing with high potential loss is different from dealing with low potential loss.

D) Software Development Life Cycle

Every period of the Software Development Life Cycle (SDLC) is helpless against various sorts of risk components. Distinguishing and understanding these risks is a preparatory stage for overseeing risks effectively. The paper introduces a complete hypothetical investigation of the real risk components debilitate each of SDLC stages. A thorough rundown of 100 risk elements was delivered. These rundown mirrors the most oftentimes happening risk figures that are regular to most programming improvement ventures [7], [8].

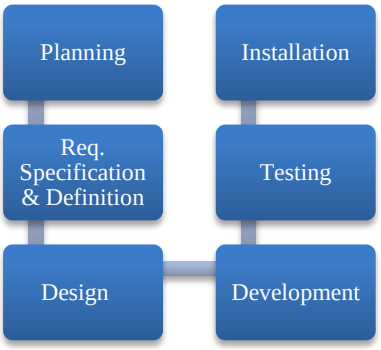

Fig. 3. Stages of SDLC

V. PROPOSED MODEL

In this exploration we checked on and chose numerous analysts identified with the Software Development and classes of the risks into high, medium and low level risks. The rundown of risks on premise of chose risks was done, which built up the poll for overview.

Summary of Pointers

From the taxonomy given the main groups of the risks category were selected for the research. The classification is given in Table 2.

Fig. 4. Process for Factors Prediction Pointers gathering

Table 1: Sorting Risk grouping

Level	Risk Group	Level	Risk Group
1	User	15	Unrealistic requirement
2	Requirement	16	Change management
3	Cost	17	Estimation
4	Schedule	18	Project management
5	Quality	19	Design
6	Business	20	Developer
7	People	21	Development
8	Project complexity	22	Process
9	Planning & control	23	Stakeholder
10	Team	24	Software
11	Organizational Environment	25	Culture
12	Scope change	26	Time dimension
13	Communication	27	Resource availability
14	Overdrawn budget		

Table 2: Category for software development

Level	Risk Group
1	Non Functional
2	Cost
3	Estimation
4	User
5	Requirement
6	Technical
7	Complex
8	Team

VI. RESULTS

A) Standardization of the Questionnaire

The specialists helped in the improvement of the survey that had the learning of programming advancement administration. The qualified specialists were chosen with up to 9 year involvement in programming improvement from understood associations.

B) Data Collection and Validation

The gathering of information was done from the specialists with the same assessment and the bearing, concur, likely concur or oppose this idea. On the off chance that one of them gave answer of not-concur than the choice differ was chosen. So the component brought on the risk for programming venture originated from the same heading of the survey.

C) Factors Extraction and Results

It is the method to separate and group the all factors from quality analysis and develop the questionnaire with the objective to select the factor with the risk and analyze from these risk factors for the prediction of the risk.

D) Application Modules

The outcome was examined by utilizing ordinal relapse of the respondents which were made out of 30 inquiries. The synopsis of the considerable number of answers of the inquiries of the respondents from various programming houses was broke down by SPSS program. The outcome and examination given by the apparatus is given beneath.

Table 3: Case Processing Summary

		N	Marginal Percentage
Satisfaction level	Extremely Unsatisfied	4	
	Unsatisfied	29	
	Satisfied	20	6.7%
	Extremely Satisfied	7	48.3%
Valid		60	33.3%
Missing		0	11.7%
Total		60	100.0%

The negligible rate records the extent of substantial perceptions found in each of the result variable's gatherings. of the 60 subjects, 4 and 29 classified as to a great degree unsatisfied and unsatisfied subsequently peripheral rate for unsatisfied reactions is 55% that demonstrates that more respondents are unsatisfied as contrast with fulfill.

Table 4: Model Fitting Information

Model	-2 Log Likelihood	Chi-Square	Df	Sig.
Intercept Only	136.470 105.978	30.492	6	.000

Link function: Logit.

The above small p-value=0.000 from the LR test, <0.05, would lead us to conclude that at least one of the regression coefficients in the model is not equal to zero.

Table 5: Goodness-of-Fit

	Chi-Square	Df	Sig.
Pearson Deviance	184.909 104.591	165 165	.138 1.000

Link function: Logit.

The above result shows that the p-value is greater than 0.05, therefore the hypothesis that model is of good fit not rejected here.

Table 6: Pseudo R-Square

Cox and Snell Nagelkerke McFadden	.398 .443 .221

Link function: Logit.

Here the Pseudo R-square (Nagelkerke= 44.3%) indicates that six independents variables explain a relatively moderate proportion of the variation in dependent variable (satisfaction levels) which is 44.3% and model is adequate enough. There may some other important predictors of Satisfaction levels that have an effect on the dependent variable.

Table 7: Parameter Estimates-I

	Estimate	Std. Error	Wald	Df
Threshold [Satisfaction = 1]	-.194	2.310	.007	1
[Satisfaction = 2]	3.702	2.350	2.481	1
[Satisfaction = 4]	6.387	2.474	6.667	1
Location Estimation Risk	.563	.391	2.068	1
Tech_Risk	.144	.394	.134	1

Table 8: Parameter Estimates-II

	95% Confidence Interval	
Sig.	Lower Bound	Upper Bound
.933	-4.721	4.333
.115	-.905	8.309
.010	1.539	11.236
.150	-.204	1.330
.714	-.628	.917

Link function: Logit.

In ordinal regression model, if there are "*k*" categories of dependent variable then a total of "*k* – 1" models are estimated. Here the dependent variable has 5 categories and one of them is not observed, of 4 there are 3 models are estimated.

The results of above table show that complex risk has a positive statistically significant regression relationship on satisfaction level.

VII. CONCLUSION

The model is perfectly suited for any evolutionary software process because it follows the same philosophy. The risk assessment and estimation steps are conducted at each evolutionary cycle with increasing knowledge and decreasing variance. The research formalizes an improvement in the evolutionary software process, introducing a risk assessment step that can be automated, and that can help shape the planning of the project in the early stages when there is still substantial freedom to allocate available time and budget. The reason for study was finding the risk which comes in software advancement and causes obstacles in fruitful programming improvement. To enhance the product procedure by expelling these risks at the early phases of software improvement helping the engineers naturally distinguishing and alleviating risks. The preparatory risk arrangement was done on the premise of risk as often as possible happens being developed stages, basic issues to enhance process for getting the proficiency amid programming improvement projects. So the expectation of risk was acquired from programming designers, it will help in basic leadership to enhance do advancement handle legitimately. The model is superbly suited for any developmental programming procedure since it takes after the same reasoning. The risk appraisal and estimation steps are directed at each developmental cycle with expanding information and diminishing fluctuation. The examination formalizes a change in the transformative programming process, presenting a risk evaluation step that can be computerized, and that can shape the arranging of the task in the early stages when there is still generous opportunity to apportion accessible time and spending plan.

REFERENCES

[1]. Bazaz, Y., S. Gupta, O. Prakash Rishi and L. Sharma. (2012). Comparative Study of Risk Assessment Models Corresponding to Risk Elements, IEEE-International Conference On Advances In Engineering, Science And Management 61-66. Journal. 213-232.

[2]. Kakkar, K. SNMC, India Shah, R. and Kakkar, M. (2013). Risk analysis in mobile application development. IEE.429 – 434.

[3]. Omni T. and J. Singh. (2013). Automated risk identification tool for software development process. Oriental Journal of Computer Science and Technology. 2320-8481

[4]. Suri,P. K. and K. Narula. (2013). Strategies for evaluating risks to project schedule. International Journal of Computer Applications. 0975 – 8887.

[5]. Ykhlef, X. and Y. Algawiaz. (2014). Defect detection and prevention Techniques based on Risk management. IEEE. 9(80): 1-4

[6]. Haneen H. , Shihadeh A. and H. Muaidi. (2014). Risk factors in software development phases.European Scientific

[7]. Soumya K. and A. V. Vidyapeetham. (2015).Software development risk aspects and success frequency on spiral and agile model. International Journal of Innovative Research in Computer and Communication Engineering. 301-310.

[8]. Toit, D.D. (2013). Political Risk Factors: What Chinese Companies Need To Assess When Investing In Africa. Stellenbosch. 1-24.

[9]. Vasile, D. and T. M. Vlăduț. (2013). Using probability – impact matrix in analysis and risk assessment. Scientific Papers.

[10]. Xie, X., C. Kothe, S. Welke and M. Rötting. (2012). Formal specification of non-functional properties of component-based software systems. Software and Systems Modeling 9(2): 161-201.

[11]. Berg, P.H. (2010). Risk Management: Producers, Methods and Experiences. IEEE Software. 1(17):79-95.

[12]. Chandan K. and D.K. Yadav (2013). A Probabilistic Software Risk Assessment and Estimation Model for Software Projects. Science Direct. 353–361.

[13]. Chen, C. (2011). Systematic Risk Management in Enterprises. Software and Systems Modeling. 7(2): 16-20.

[14]. Dr. Ajeet S. P. (2014). Recent Advances In Risk Analysis And Management (RAM). COMPUSOFT, An international journal of the advanced computer technology. 1412-1416.