

MRMM: A Mathematical Risk Management Model for Iterative IT Projects based on the Smart Database

A. S.Khatavakhotan, N. HashemiTaba, and S. H.Ow

Abstract—The increasing advancement in information technology industry is compatible with the changing nature of the world. While software is important for all facets of the modern world, software development itself is not a perfect process. Iterative software development methods have recently emerged as a new and better way of developing software as compared to traditional methodologies. These methods are developed to make prompt changes in a constantly changing world. The more modern the technology and the methods are, the newer the risks that must be properly controlled and managed. Therefore, focusing on risk management process in IT projects will lead to applicable results. Due to the high failure rate in IT projects, increasing the focus on risk management seems essential.

Index Terms—Budget Estimation, Iterative IT Projects, Risk Management, Smart Database

I. INTRODUCTION

The management of risk considered as a key discipline for project management [1]. According to the latest research done in 2007, 32% of software projects fail [2]. Failing a project means not implementing the planned program. The risk management role is vital to decrease the failure rate. Although, different risk management methods are argued in the past years, but still there is not a formal reliable model for risk management [3]. Especially in Iterative projects risk management must be done promptly and accurately to avoid rework. The lack of quality of large software likely decreases the total performance of a company[4].

In this paper the main issue is to present a Mathematical Risk Management Model for iterative IT projects (MRMM). The proposed mathematical model benefits from a smart database in final stage. The spectrum of activities covers a wide scope from risk identification to risk budget estimation and allocation. The assumed iterative software development model in this research is Agile iterative model. The aim is to explore the existing risks of the Agile method and present an appropriate model for a comprehensive rapid risk management plan.

II. THE MRMM MODEL

The proposed MRMM model is designed based on the Agile software development methodology. Many of

scholars have focused on software error to mitigate the software risk instead of software development risks [5]. To compensate this weakness, the presented model in this paper is considered both project management and development process aspects. The main point of the iterative Agile methods is responding to change [6]. As the software development process becomes faster by Agile approach, risk management process must be adapted with that. Therefore, all phases of risk management should be done completely but promptly in and Agile process [7]. Agile approach is more human based rather than process based [8]. Therefore, the major class of software risks refers to people behavior in risk management process.

III. THE STEPS OF THE MRMM MODEL

Figure 1 shows the main steps of the proposed model. The first step starts from risk identification and the last step focuses on the result analysis and interpretation.

Step 1- Identifying the risks by categorizing them.

Step 2- Analyzing each risk in any iteration to determine:

- 1) Which tasks are affected by the mentioned risk
- 2) Computing the weight of risk by:
 - How relevant each task is (low to high)
 - If removal of the risk is necessary for system accuracy based on historical data
- 3) Calculating the Risk Factors amount (F1)
- 4) Prioritizing the known risks based on the Biased Risk factors (F2)

Step 3- Obtaining the cost of each risk

Step 4- Calculating the total risk amount (F3)

Step 5- Calibrating of total risk according to the used iterative Agile method

Step 6- Computing the requested risk budget (F4)

Step 7- Calculating the real budget for each risk (F5)

Step 8- Monitoring the risk removal process

Step 9- Analyzing the acquired results in order to apply in the next iteration

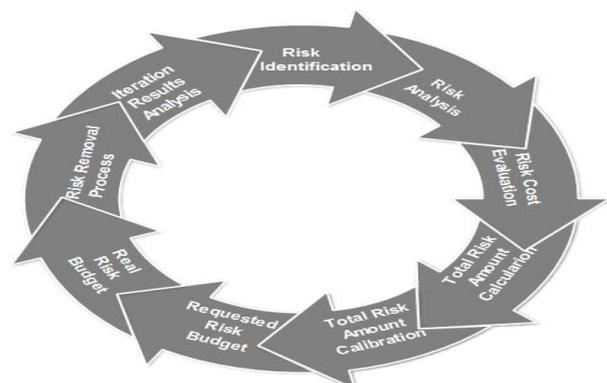


Fig. 1. The MRMM model steps

Manuscript received September 19, 2011; revised October 25, 2011.

The authors are with the Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya, 50603, Kuala Lumpur, Malaysia

IV. THE EQUATIONS OF THE MODEL

In step 1, risk identification will be completed and a preliminary list of potential risks based on previous iterations (except for the first iteration) will be provided. This is inspired by the historical data meaning [9]. Risk identification is a difficult task in every risk management methodology [10].

Step 2 has four stages. Stage I, logs the affected tasks by the aforementioned risks in the preliminary list, while the second stage (II) considers two important features for any of the risks, relevancy and removal necessity. These features facilitate the risk identification to prioritize the significant risks and update the initial list. By entering the stage III the Eq. (1) helps to calculate the risk factor amount. Step 2 finishes by prioritizing risks by biased risk factors Eq. (2).

A. Risk Factor and Total Risk Amount Calculation

Each ‘Risk Factor’ has a definition and a value. $h_{(i,k,ss)}$ shows the amount of rework hours (overtime) for risk i , to be performed by semi-skilled person in iteration j (k is the counter variable for j). $H_{(k,ss)}$ is the total amount of rework hours (overtime) for all risks, to be done by semi-skilled people in iteration j for Risk i , Iteration j : (1)

$$RiskFactor_{i,j} = \frac{\sum_{k=1}^{j-1} k \times [h_{(i,k,ss)} \times N_{(k,ss)} + h_{(i,k,s)} \times N_{(k,s)} + h_{(i,k,hs)} \times N_{(k,hs)}]}{\sum_{k=1}^{j-1} k \times [H_{(k,ss)} \times N_{(k,ss)} + H_{(k,s)} \times N_{(k,s)} + H_{(k,hs)} \times N_{(k,hs)}]}$$

where N is a ‘Normalized Coefficient’, it is calculated according to the ranking of involved people in the needed hours (high-skilled, skilled, and semi-skilled). The ‘Normalized Coefficient’, which is mentioned above, is used in the method just for first iteration and have to be calibrated based on the historical and the environmental data that formed the following metrics for the other iterations. The meaning of metrics and ‘Calibrated Normalized Coefficient’ which is shown by ‘ n ’ symbol are presented in next section:

$$N = \begin{cases} 1 & semi_skilled(ss) \\ 2 & skilled(s) \\ 3 & high_skilled(hs) \end{cases} \quad (2)$$

For each identified risk ‘ i ’, in iteration ‘ j ’, the criteria of RiskFactor calculation will be as follows: For the first iteration, the sums will be used just based on the experience had been acquired from similar projects. Therefore, risk enacting in this model will be from the second iteration.

To calculate risk factors in iteration (n), all risks in first iterations to the iteration ($n-1$) will be considered. Just the only difference, is that like ‘COCOMO’ estimation model, which was presented by Boehm, the last iterations will have more weight than the first ones. Hence, risk management normalized hours in each iteration, will be multiplied by the number of that iteration and in this way will gain weight in accordance with the closeness of that iteration to the current iteration. The variable is used for the display of iterations (the weight of each iteration is K).

B. Metrics and its Usage for the Calculation of ‘ n ’

The metric used in this paper is based on the number of lines of code (LOC) produced by each category of people. Hence, the ratio of the number of lines written by high-

skilled, skilled, and semi-skilled to the total number of lines will be normalized quotient (calibrated N). This amount is obtained on average, for three categories of high-skilled, skilled, and semi-skilled people, as the following equations:

$$\begin{matrix} j\text{-iteration} \\ \frac{LOC_{j,hs}}{LOC_{j,total}} \times \frac{NOP_{j,hs}}{NOP_{total}} = n_{j,hs} \end{matrix} \quad (3a)$$

$$\begin{matrix} j\text{-iteration} \\ \frac{LOC_{j,s}}{LOC_{j,total}} \times \frac{NOP_{j,s}}{NOP_{total}} = n_{j,s} \end{matrix} \quad (3b)$$

$$\begin{matrix} j\text{-iteration} \\ \frac{LOC_{j,ss}}{LOC_{j,total}} \times \frac{NOP_{j,ss}}{NOP_{total}} = n_{j,ss} \end{matrix} \quad (3c)$$

where NOP_{total} is the total ‘Number Of People’ involved in process (hs, s, ss), also $NOP_{j,hs}$ means the number of high-skilled people in iteration j .

$$LOC(j,hs) \approx N(j,hs) \quad (4a)$$

$$LOC(j,s) \approx N(j,s) \quad (4b)$$

$$LOC(j,ss) \approx N(j,ss) \quad (4c)$$

C. Calculation of weight Function and Biased Risk Factor

Risk calculation in a historical stream is the amount of Risk Factors in every running or development of a system that is multiplied by relevancy and necessity factors. In a history based model of Cheng and Jiang the only factor which is considered is relevancy; however, in the suggested model here, the weight of any factor w , is a function of two variables n , r (necessity and relevancy both)[9]. To make it simpler, this function is defined as discrete function between 0 and 2.

$$w(r,n) = \begin{cases} 2 & \dots \dots \dots r = 1, n = 1 \\ 1 & \dots \dots \dots r = 1, n = 0 \\ 0 & \dots \dots \dots r = 0, n = 0 \end{cases} \quad (5)$$

$$\begin{matrix} Risk : i - Iteration : j \\ BiasedRiskFactor_{j,i} = w_i(r_j, n_j) \times RiskFactor_{j,i} \end{matrix} \quad (6)$$

$$TotalRiskAmount_j = \sum BiasedRiskFactor_I \quad (7)$$

D. Risk Requested Budget and Real Cost Estimation

After accomplishing the risk identification and analysis stages, the risk budget request and the allocation policy would be initiated.

$$RequestedRiskBudget_j = RiskCost_j \quad (8)$$

$$RiskCost_j = Expected_Cost_j \times TotalRiskAmount_j \quad (9)$$

$$Expected_Cost_j = H_{ss} \times Fee_{ss} + H_s \times Fee_s + H_{rs} \times Fee_{rs} \quad (10)$$

Fee is the payment for each hour of human work. For more accuracy in this equation, considering the cost based on ranking of people is important.

For risk management, first the sum of total needed budget and costs of each risk is calculated. The budget can be computed by the following equation:

$$Requested\ Risk\ Budget = \sum_{i \in E_{total}} RiskCost_i \quad (11)$$

The risk management has to be applied based on the assigned budget, not the requested budget, because, these two budgets are not often equal. The real allocated (given) budget to any risk i could be calculated by the following equation: (12)

$$BudgetRisk_{j,i} = \frac{BiasedRiskFactor_{j,i}}{TotalRiskAmount_j} \times AssignedRiskBudget_j \quad (12)$$

V. A DETAIL EXPLANATION OF THE PROPOSED MATHEMATICAL MODEL

The point that categorizing programmers as high-skilled, skilled and semi-skilled is a qualitative definition. In order to have survey on models, or even define them, quantitative definitions are needed. For instance, if a programmer can code 10,000 error-free lines, he is a professional in this way. We can say 5000 lines of programming with the same tool in the same environment, entitles this programmer as a semi-skilled and 2000 lines with the same tool and the same environment as a beginner. On the other hand, we can find out that a high-skilled programmer is able to code a program, which cannot be written from technical point of view by a semi-skilled.

Risk identifying, categorizing, and handling, finding their factors, weighting known factors, making a diagnosis if they are relevant, irrelevant, required, or useless in one iteration is essential. All of these activities will make risk management steps of MRMM.

Risk management in iterations of Agile methods, which is management point of view, can be both a challenge and an opportunity. In an iteration, which takes some weeks is hard, costly, and even impossible (if has no automated tool or appropriate model) that risk identifications to be done. However, since all similar activities are performed in successive iterations, therefore, similar risks will do exist, which will be controlled and managed well as time goes on. Then the aforementioned challenge will convert into a management opportunity.

Another important point, the weight for calculating "RiskFactor" has two variables. The first clarifies the relevance of "RiskFactor" with the current situation and the second parameter shows the necessity of control and the management of the mentioned factor. In order to have a better understanding, it is better to note that Agile methods mainly recommend face-to-face conversation with clients. This measure is taken for two purposes. First,

determination of the requirements and second rating each prepared version of product. Therefore, from risk management standpoint, if a discovered error needs to be repaired or reworked, it is considered an occurred risk. It may be deemed neglectable by the client and in such conditions necessity parameter is zero and the weight of RiskFactor will be so low. The second states which a supplement description is, when an irrelevant danger seems essential for project process, and its objectives. As an example, we can mention the consideration to do with user interface. In the implementation of software system, which has hundreds of functions and operations, sometimes the physical placement of a field on screen, foreground or background (and having no adaptation with the user's taste) cannot be a relevant task with main objectives, but since end-users confirmation is the condition of project approval, more weight will be given to weight determination of the mentioned RiskFactor. In order to highlight paying attention to risk weight, weight function is defined in MRMM model.

VI. SMART DATABASE AS A CASE TOOL

Main features and potential of risk case tools' software for Agile method are mentioned in this section. CASE (Computer Aided Software Engineering) tools are very important in software engineering. By their application, not only software development is accelerated, but also the number of errors minimizes. In a way, we have harmony in design and implementation. Some case tools are just considered to automate different parts of a software engineering process. Two recommended CASE tools are mentioned below: To analyze risk management of Agile methods, it is essential to have an automated tool based on relational DBMS to log the risk information of iterations. Another specific CASE tool is Agile Risk Information Sheet (ARIS). The information fields of the risk in each task are stored as one record of any iteration (See figure 2). In the same iterations in multiple records, risk changes will be kept which has a significant difference with presented RIS in Pressman Software Engineering book, so it is called "Agile RIS" or "ARIS". ARIS means logging and registering information, which are related with different risks in an Agile iteration. The designed ARIS is called smart database in this research, as the information of each iteration logs on each of the aforementioned database and will transfer to the next iterations.

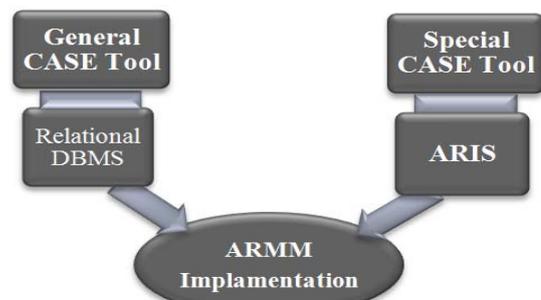


Fig. 2. MRMM recommended CASE tools

By logging this information in next iterations, the following results are obtained: a) pay close attention to previous iterations and predict accordingly necessary facilities and equipment; b) determine the weight of risk

factors according to the history of effective risks in previous iterations (from the first iteration to the current iteration). ARIS is depicted in Figure 3.

Agile Risk Information Sheet (ARIS)				
Project name:		No. of Iteration:		
Risk ID :	Date: / /	Category	Activated date: from / / to / /	Record No.
	Time: : min			
Description:				
High skilled person hours:	Normalized Coefficient (hs):			
Skilled person hours :	Normalized Coefficient (s) :	Total Hours		
Semi-killed person hours :	Normalized Coefficient (ss):			
Risk factor amount (considering previous Iterations):				
Affects on :	Chunk no.1	Planned hours:	Overtimes:	Total hours:
		Reasons:		
	Chunk no. n	Planned hours:	Overtimes:	Total hours:
		Reasons:		
Mitigation activities:				
Monitoring activities:				
Contingency Plan:				
Current Status:				
Involved People's ID:				
Comment:		Name , Id and Signature :		

Fig. 3. Smart Database used in MRMM process

VII. RISK MITIGATION, MONITORING AND MANAGEMENT

TABLE 1 .MRMM IN AGILE RISK MANAGEMENT METHOD

Task (MRMM)	Responsibility
Risk Mitigation	Logged information in ARIS records about the reason of the occurrence of the dangers can be used in the programming of similar cases done.
Risk Monitoring	In each iteration, the daily survey of to what extent-planned hours are in accordance with real Hours, which is possible by logged information in ARIS is highly demanded.
Risk Management	In each iteration, the daily survey of to what extent-planned hours are in accordance with real Hours, which is possible by logged information in ARIS is highly demanded.

Risk management can be applied properly in the project, when enough information of previous status of project is available for risk manager. Mitigation and monitoring are both supplementary tasks to do with management to control and decrease the possibility of future risks. An overview of MRMM is presented in Table 1.

VIII. CASE STUDY

In order to verify the model, a case study has been completed. This case study has depicted accurate risk costs and estimation of an IT company which produces web-based and windows-based software for oil and gas industry [11]. The relevant information by one of the Agile methods (Scrum) in two years is gathered from this company and applied to the proposed mathematical model. According to the real available data six related risks were found. These risks are as follows:

- Product managers located remotely
- Lack of experience in creation user stories
- Technology issues such data access
- User and service documents may not be translated into proper languages
- Requirements currently unclear
- Some team members not assigned full-time

The related tasks and mitigation activities based on the designed smart database analyzed. Finally,comprehensive risk management plan including budget estimation and allocation is produced successfully with details for the aforementioned company.The results interpretation sll be presented in the future study.

IX. SUMMARY

Agile methods are almost newfound in IT era. Therefore, there are unknown risks in this emerging method. The reason is the rapid iterations in Agile approach. Software risk not only affects the software projects outcomes, but also generates crucial failures in final product.The Agile development process is faster than conventional models, which makes the risk management more difficult to be applied. In an iterative Agile approach the process risks occur and repeat and intensify gradually. The impacts will be amplified promptly.

This paper presented a novel model for risk management. Nevertheless, this theoretical model would not be sufficient and risk management process is not applicable unless the risk costs and estimation complete. Therefore, correspondent risk budget estimation equations are added to make the model complete. By completing each iteration, the historical data become more accurate and real. Finally, to verify the model a case study of an IT company with Agile infrastructure examined the correctness and accuracy of the equations.

Two general and specific recommended CASE tools facilitate the model implementation. Using a relational DBMS helps the developers to record and retrieve the necessary risk information and the ARIS database is a smart assistant for using the historical related data from previous iterations in current interaction of the development process.

REFERENCES

- [1] E. Kutsch, and M. Hall, "The rational choice of not applying project risk management in information technology projects," *Project Management Journal*, vol. 40, no. 3, pp. 72-81, 2009.
- [2] R. Wienclaw, "Information System Project Management," *Information System Project Management -- Research Starters Business* [serial online]. June 2008, Available from: Research Starters - Business, Ipswich, MA.
- [3] S. Yuwei, "Today's Solution and Tomorrow's Problem: The BusinessProcessOutsourcing Risk Management Puzzle," *California Management Review*, vol. 49, no. 3, pp. 27-44.
- [4] T. Abdullah, A. Mateen, A. Sattar, T. Mustafa, "Risk Analysis of Various Phases of Software Development Models," *European Journal of Scientific Research* [serial online]. February 15, 2010; vol. 40, no. 3, pp. 369-376.
- [5] Peng Y, Kou G, Wang G, Wang H, Ko, F. Empirical Evaluation of Classifiers for Software Risk Management. *International Journal of Information Technology & Decision Making* [serial online]. December 2009;8(4):749-767.
- [6] B. Kent, et al. "Manifesto for The agile Software Development," www.theagilemanifesto.org, February 2001.
- [7] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of Migrating to The Agile Methodologies," *Communications of the ACM*, vol. 48, no.5, pp. 73-78, 2005.
- [8] S. Pichler, and R. Pichler, "The Agile Risks," *The Agile Rewards Software Development*. San Francisco: Vol.13, Issue 4, pp. 50-53, 2005.
- [9] H. Jiang, C. Chang, X. Jinchun, and S. Cheng, "A History-Based Automatic Scheduling Model for Personnel Risk Management," *Compsac*, pp. 361-366, 31st Ann. Int. Computer Software and App. Conference - Vol. 2, 2007.
- [10] N. Petria, and L. Petria, "Operational Risk Management and Basel II," *Revista Academiei Fortelor Terestre* [serial online]. December 2009, vol. 14, no. 4, pp. 96-100.
- [11] C. Richard Mann, "An Exploratory Longitudinal Case Study of Agile Methods in a Software Company," Department of Computer Science, University of Calgary, Alberta, Canada, August 2005, ISBN 0-494-05471-9.