Entropy-based Model for Measuring Risk of Requirement Changes

Yu Yang*, Hua Zhou, Junhui Liu
School of Software
Yunnan University
Kunming, 650000, China
e-mail:yuwenyunying@gmail.com

Yun Feng
Department of Computer and Information Science
Zunyi Normal College
Zunyi, 563000, China
e-mail:fyun3ok@163.com

***Abstract*—Software projects face a common problem is the requirement uncertainty and frequent requirement changes. The risk of requirement changes is considerable risk in software project management. Information entropy can effectively measure subsystem's degree of uniformity. This paper proposes a quantitative risk measurement model that could be used to measure risk posed by requirement changes. The more uniform requirement impact on software project, the smaller the risk, otherwise the key requirement changes will have a significant impact on the project. This paper discusses the rationality of the model and gives an instance of this model. The model requires data can be obtained from enterprise. Experiments show that the model is scientific and rational. It can serve as a reference for requirement management.**

Keywords- risk measure; information entropy; requirement change; software engineering

I. INTRODUCTION

Good requirement management is crucial to reduce development costs and guarantee success of project [1]. According to authoritative statistics, in the world wide only 25% of the software development projects could achieve client's goal in the stipulated time and budget [2]. So it is difficult for developers to develop a reliable system that meet user's needs in the stipulated time and budget. Software projects face a common problem is the requirement uncertainty and frequent requirement changes[3]. Frequent requirement changes will inevitably affect the progress, cost and quality of the project. Information entropy is an important theory in System Science; it could be used to measure the degree of uncertainty and uniformity. This paper proposes a quantitative risk measurement model that could be used to measure risk posed by requirement changes. The experimental results show that the model can effectively measure the risks of requirement changes.

II. THE DEFINITION AND ANALYSIS OF SOFTWARE PROJECT RISK

Software project risk could definite as uncertain events arise in the process of software development. It may result in lowering the quality of software program even cause financial losses for IT enterprise [4]. Software project risk management is an important content of software project management. In the process of software project risk management, firstly identify risk, secondly assess the probability that they appear and influence, and at last build a plan to manage risk.

The main goal of risk management is risk prevention. We can minimize the risk if we do project risk management. However, current domestic IT enterprises do not pay much of attention to risk management of software project, which will result in software project delay, over budget, or even failure. Successful project management is generally with a good project risk management. Therefore, any system development projects should treat risk management as an important content in software project management. Common software project risks are as shown below: product orientation errors, staff turnover, project management failure, development goals are not clear or wavering, Implementation of development plan is severely affected, technical solutions are flawed, project cost overrun, the development environment and process management confusion, poor product quality, frequent requirement changes and so on.

This paper is only focus on researching the risk of software requirement changes. Requirement changes are often expressed as a recurrent and high-impact risk in all forms. Requirement change is inevitable and will run through the entire cycle of the project development. Requirement change is full of considerable uncertainty.

In an actual software project, we classify requirement change into two types: the first one is changes under the premise of requirements are not very clear. Usually because clients do not have enough capability or time to define requirements in the process of defining requirements or defined requirements are not detailed, so clients often ask for requirement changes after requirements confirmation. We call this kind of requirement change as missing requirement change. The second is changes arise under the premise of requirements are clear, usually because the current needs of business and originally proposed needs has changed or the original function is no longer applicable, We call this kind of requirement change as additional requirement change.

We can generally divide requirements change impact on software projects into three cases:

- Affect the project's overall product quality.
- Affect the project completed within budget.
- Affect the project completed within the expected progress.

Although missing requirement change or additional requirement change are proposed for positive development

978-1-4799-5245-8/14 $31.00 © 2014 IEEE
DOI 10.1109/ISCC-C.2013.70

of the system, their impact on the software project is extremely hazardous.

III. Rationality Analysis of Measuring Requirement Changes Risk with Entropy

The concept of entropy originated in physics, it is used to measure the degree of disorder of a thermodynamic system. In information theory, entropy is a measure of the uncertainty in a random variable [5]. In this context, the term usually refers to the Shannon entropy, which quantifies the expected value of the information contained in a message. In this context, a 'message' means a specific realization of the random variable. This definition of "entropy" was introduced by Claude E. Shannon in his 1948 paper "A Mathematical Theory of Communication". Entropy is the quantitative measure of disorder in a system [6]. For a random variable X with n outcomes $\{x_1 \ldots\ldots x_n\}$, a measure of uncertainty is denoted by H(X), is defined as:

$$\mathrm{H(X)} = \sum_{i=1}^{n} p(x_i) I(x_i) = -\sum_{i=1}^{n} p(x_i) \log_b P(x_i)$$

$p(x_i)$ is the probability mass function of outcome. b is the base of logarithm, the value of b is usually 2 or the constant e or 10. When b = 2, the unit of it is bit. When b = e, the unit of it is nat. When b = 10, the unit of it is dit.

Nature of information entropy:

- Symmetrical characteristic: although the sequence of events in the probability system are different, the entropy value of probability system remains unchanged, which means entropy value of probability system has nothing to do with the sequence of events.
- Non-negative characteristic: entropy is a non-negative value. Because all values of the random variable X is the probability distribution of 0 <P <1
- Determinacy: If an event in a probability system is inevitable, other events cannot occur, the value of entropy is 0.
- Extremum characteristic: The entropy value is maximum when the event in the system is a uniform probability distribution

The greater the uncertainty of variables, the greater the entropy, the greater the amount of information needed to understand it. Information entropy is a concept used to measure the amount of information in the field of information theory. The more ordered a system, the lower information entropy of it. Conversely, the more disordered a system, the higher information entropy of it. So we could consider entropy as a measure of degree of ordering for a system.

Software project is a system of dissipative structure. It exchange largely with outside, such as information exchange. It presents a non-equilibrium state. Software project is a system of dissipative structure. It exchange largely with outside, such as information exchange. It presents a non-equilibrium state. Various elements within a software project affect each other in a complex non-linear way, and there is a fluctuation phenomenon in it. The fluctuation (change) of software project requirement could cause the change of system entropy; it can ultimately cause the fluctuation of system. But not all of the fluctuations could induce the system to change. Small fluctuation will be dissipated by the system, only larger fluctuations could induce system to change, that is bring great risk to software project. Therefore, the measure of software project risk is the measure of the uniformity degree of requirements impact on the project.

IV. Entropy-Based Model for Measuring Risk

Risk analysis of requirement change has great influence on the success of the project, so a lot of scholars have put forward the idea of risk management. There are some qualitative risk analysis methods in actual application, in which the most representative are the following:

(1) Samson improved the definition of software requirements by researching potential problems in requirement analysis. [7] He divided the problems of requirement analysis into 4 types: not tight, contradiction, performance target is not clear and lack of knowledge in some field.

(2) Robinson put forward Risk Assessment of Requirements Engineering (RARE) to identify nine types of requirements errors and evaluate requirement risk[8]. We are able to identify 9 types of requirements errors and risk of requirements evaluation with RARE. RARE takes into account 9 kinds of common requirements errors, including conflicts, domain independent, complementary incomplete, inconsistent, lack of understanding, vague, imprecise, grammatical complexity and redundancy. RARE provides a definition of requirements errors, a testing of requirements errors, a classification of requirements errors and error assessment. RARE assesses the level of requirement risk (high, medium, low), and defines the level of risk with appropriate error class. RARE set of requirements as input the output of the total risk assessment .This method is used to decide whether to continue or discontinue requirements activities or directly step into design phase.

(3) Armour pointed out that requirement change includes changes that should be prototyped, he also pointed out that those requirement changes or potential requirement changes have higher risk[9]. The cause of requirements volatility is that users couldn't describe requirements accurately. The reason may be due to changes in technology, system development cycle is long or immature requirements gathering and so on. Armour said it will lead to requirements volatility if do not pay a return visit to the requirements that gathered in the requirement analysis of early stage. He recommended the establishment of prototype of these requirements volatility.

(4) Myers put forward a method to help project managers to identify early the causes of requirement risk in the project life cycle by setting the priority of software requirements[10].

(5) Romano pointed out the problem that how to identify potential requirement risk. He put forward a structured method to identify requirement risk for auxiliary decision

system. The evidence that risk appears or does not appear comes from inspection of related risk factors that involve requirement item. Romano proposed a method named Test-Based Risk Identification Methodology, which can be used to detect 4 types of main potential requirement risk: ambiguity, conflict, complexity, technical factors. In his TBRIM method, all requirements should be tested according to guidelines, so people could determine whether the problem exists and whether it needs to be corrected[11].

(6) Donald put forward a Volatility of Requirements Assessment Method (VRAM). This is a structured method for auxiliary decision system [12]. VRAM evaluate volatility according to history of application requirements and requirement type. VRAM get volatility factor from actual data and evaluate all kinds of requirement volatility through Analytic Hierarchy Process (AHP). This method is used to evaluate the level of requirement volatility for a software project. This method also evaluates the potential impacts caused by new technology, users' requirement changes and incorrect assignment for requirements.

The above methods are only a definition or a qualitative analysis of the risk caused by requirement change, and do not give a quantitative risk value, this paper puts forward a quantified method for measuring risk of requirement changes. It can effectively quantify the value of risk caused by requirement change.

We could define a three-tuples R= (S, P, C) for the risk of requirement changes. S means the requirement of software project, P means the probability of risk occurrence, and C means loss caused by risk. We presume there are n requirements in a software project, we set them as: S_1, S_2……S_n . Impacts caused by missing requirement change are: C_1, C_2……C_n. The probability of occurrence of missing requirement change is: P_1, P_2……P_n. Impacts caused by additional requirement change are: C_{11}, C_{12}……C_{1n}. The probability of occurrence of missing requirement change is: P_{11}, P_{12}……P_{1n}; Impacts caused by additional requirement change are: C_{21}, C_{22}……C_{2n}. The probability of occurrence of additional requirement change is: P_{21}, P_{22}……P_{2n}; The values of degree of requirements change influence on the project is: A_1, A_2……A_n

$$A_i = C_{1i} \times P_{1i} + C_{2i} \times P_{2i} \qquad i = 1,2\ldots\ldots n \qquad (1)$$

We present normalization to deal with A_i then we get:

$$\rho_i = \frac{A_i}{\sum_{i=1}^{n} A_i} = \frac{C_{1i}\times P_{1i}+C_{2i}\times P_{2i}}{\sum_{i=1}^{n}(C_{1i}\times P_{1i}+C_{2i}\times P_{2i})} \qquad i = 1,2\ldots\ldots n \qquad (2)$$

From the theory of information entropy, we know:

$$H = -\sum_{i=1}^{n} \rho_i \log_b \rho_i \qquad i = 1,2\ldots\ldots n \qquad (3)$$

According to the nature of entropy, the more uniform ρ_i is, the higher the value of entropy (H) is. When $\rho_1 = \rho_2 = \cdots = \rho_n = 1/n$, we can get the highest value $H = \log_b n$. But the value of Equation (1) may be greater than 1. To facilitate measurement, we present normalization to deal with H, we get:

$$H^* = \frac{H}{H_{max}} = \frac{H}{\log_b n} \qquad (4)$$

$0 \le H^* \le 1$, the more uniform ρ_i is, the higher value entropy H is, the lower risk is. So we can get:

$$R = 1 - H^* = 1 - \frac{H}{H_{max}} = 1 + \frac{\sum_{i=1}^{n} \rho_i \log_b \rho_i}{\log_b n} \qquad (5)$$

$0 \le R \le 1$, According to the definition of risk and information entropy theory, we can get entropy-based Model for measuring risk:

$$R = 1 + \frac{\sum_{i=1}^{n} \frac{C_{1i}\times P_{1i}+C_{2i}\times P_{2i}}{\sum_{i=1}^{n}(C_{1i}\times P_{1i}+C_{2i}\times P_{2i})} \log_b \frac{C_{1i}\times P_{1i}+C_{2i}\times P_{2i}}{\sum_{i=1}^{n}(C_{1i}\times P_{1i}+C_{2i}\times P_{2i})}}{\log_b n} \qquad (6)$$

The more uniform the value (A) of impacts caused by requirement change is, the higher value of entropy (H^*) is, the lower risk (R) is. If a requirement change is less impact on the whole software project, system can consume this fluctuation. Otherwise the higher the risk is. So we can use the model to measure the risk of software requirement change.

V. CASE STUDY

Project X comes from research management information system (MIS) of a small university. The main function of this MIS is to manage achievements in scientific research (such as research award, patent, scientific research work and so on) efficiently within the university's teachers. Teachers of this university can report their achievements in scientific research to academy, then college scientific secretary audit these achievements, achievements will be reported to department of research once they are through the audit. If department of research approve these reported achievements, these reported achievements take effect as from that time. School leaders can view and count achievements in scientific research within the school. Budget for the project is one hundred thousand Yuan. We sorted out the following modules: management of project type, inquiry of research project, management of research project, management of research financial billing, management of patent applications, management of patent license, management of software registration, management of software license, management of research papers, management of research works, management of research incentive, inquiry of scientific research situation of researchers .According to customer's requirements, we sorted out the requirements of roughly 50 function points. According to the importance of requirements, we classify requirements into 4 types:

The first one is very critical requirement. It means the requirement is critical to the entire project. These requirements often direct impact on the speed, reliability,

expandability of system. These requirements define the core essence of software. They are mainly based on the principle in program design and combine with software task requirement to define data structure and management mechanisms. It is a most important work to confirm very critical requirement at first.

The second one is critical requirement. These requirements are numerous problems and needs in the characteristics section.

The third one is general requirement. It is a set of basic operation to complete the task. These basic requirements are common part that one of many problems and needs abstract. They are basics for other functions.

The last one is minor requirement. These requirements are minor and do not need to hurry to fulfill in the development phase.

If a very critical requirement has changed as missing requirement change, it will result in the loss of 15000 additional. The probability of missing requirement change is 10% in very critical requirement. If a very critical requirement has changed as additional requirement change, it will result in the loss of 10000 Yuan. The probability of additional requirement change is 20% in very critical requirement. Very critical requirements in the project X such as: the system can 7 × 24 hours of continuous operation; the page response time cannot be more than six seconds.

If a critical requirement has changed as missing requirement change, it will result in the loss of 8000 Yuan. The probability of missing requirement change is 20% in critical requirement. If a critical requirement has changed as additional requirement change, it will result in the loss of 4000 Yuan. The probability of additional requirement change is 30% in critical requirement. Critical requirements in the project X such as: authorize system functions according to user group or role.

If a general requirement has changed as missing requirement change, it will result in the loss of 2000 Yuan. The probability of missing requirement change is 40% in critical requirement. If a general requirement has changed as additional requirement change, it will result in the loss of 500 Yuan. The probability of additional requirement change is 50% in critical requirement. Genera requirements in the project X such as: count achievements in scientific research, manage financial receipt of research project.

If a minor requirement has changed as missing requirement change, it will result in the loss of 400 Yuan. The probability of missing requirement change is 80% in critical requirement. If a minor requirement has changed as additional requirement change, it will result in the loss of 50 Yuan. The probability of additional requirement change is 60 % in critical requirement. Minor requirements in the project X such as: page fonts can be adjusted, background picture can be changed.

There are 8 very critical requirements, 13 critical requirements, 23 general requirements and 6 minor requirements. The cost of impact of requirement change are as shown in Tab. 1.We presume that the same type of requirement's probability of missing requirement change and additional requirement change is equal. More details are as shown in Tab. 2.

TABLE I. LOSS OF ALL TYPES OF REQUIREMENT CHANGE

NO	Type	Effects caused by missing requirement change	Effects caused by additional requirement change	Count
1	very critical requirement	15000	10000	6
2	critical requirement	8000	4000	8
3	general requirement	2000	500	24
4	minor requirement	400	50	12

TABLE II. PROBABILITY OF ALL TYPES OF REQUIREMENT CHANGE

NO	Type	probability of missing requirement change	probability of additional requirement change
1	very critical requirement	0.1	0.2
2	critical requirement	0.2	0.3
3	general requirement	0.4	0.5
4	minor requirement	0.8	0.6

From above tables, we can know:

$C_{11}=C_{12}=\ldots\ldots=C_{16}=15000$, $C_{21}=C_{22}=\ldots\ldots=C_{26}=10000$,
$P_{11}=P_{12}=\ldots\ldots=P_{16}=0.1$, $P_{21}=P_{22}=\ldots\ldots=P_{26}=0.2$;
$C_{17}=C_{18}=\ldots\ldots=C_{114}=8000$, $C_{27}=C_{28}=\ldots\ldots=C_{214}=4000$,
$p_{17}=p_{18}=\ldots\ldots=p_{114}=0.2$, $P_{27}=P_{28}=\ldots\ldots=P_{214}=0.3$;
$C_{115}=C_{116}=\ldots\ldots=C_{138}=2000$, $C_{215}=C_{216}=\ldots\ldots=C_{238}=500$,
$P_{115}=P_{116}=\ldots\ldots=P_{138}=0.4$, $P_{215}=P_{216}=\ldots\ldots=P_{238}=0.5$;
$C_{139}=C_{140}=\ldots\ldots=C_{150}=400$, $C_{239}=C_{240}=\ldots\ldots=C_{250}=50$,
$P_{139}=P_{140}=\ldots\ldots=P_{150}=0.8$, $P_{239}=P_{240}=\ldots\ldots=P_{250}=0.6$.

According to Equation (1), we can compute:

$A_1 = C_{11} \times P_{11} + C_{21} \times P_{21} = 15000 * 0.1 + 10000 * 0.2 = 3500$
$A_2 = C_{12} \times P_{12} + C_{22} \times P_{22} = 8000 * 0.2 + 4000 * 0.3 = 2800$
$A_3 = C_{13} \times P_{13} + C_{23} \times P_{23} = 2000 * 0.4 + 500 * 0.5 = 1050$
$A_4 = C_{14} \times P_{14} + C_{24} \times P_{24} = 400 * 0.8 + 50 * 0.6 = 350$

According to Equation (2), we can compute:

$\sum_{i=1}^{6}(C_{1i} \times P_{1i} + C_{2i} \times P_{2i}) = 21000$
$\sum_{i=7}^{14}(C_{1i} \times P_{1i} + C_{2i} \times P_{2i}) = 22400$
$\sum_{i=15}^{38}(C_{1i} \times P_{1i} + C_{2i} \times P_{2i}) = 25200$
$\sum_{i=39}^{50}(C_{1i} \times P_{1i} + C_{2i} \times P_{2i}) = 4200$
$\rho_1 = \frac{21000}{72800} = 0.2884$, $\rho_2 = \frac{22400}{72800} = 0.3077$
$\rho_3 = \frac{25200}{72800} = 0.3462$, $\rho_4 = \frac{4200}{72800} = 0.0577$

According to Equation (3), we can compute:

$H = -(0.2884 * \log_2 0.2884 + 0.3077 * \log_2 0.3077 + 0.3462 * \log_2 0.3462 + 0.0577 * \log_2 0.0577) = 1.8078$

According to Equation (4), we can get $H_{max} = \log_2 50 = 5.6439$, and $H^* = H / H_{max} = 0.3203$

According to Equation (6), we can compute the result

$R = 1 - H^* = 1 - 0.3203 = 0.6797$.

We add a very critical requirement to the project, the remaining conditions remain unchanged, and we get:

$C_{11}=C_{12}=\ldots\ldots=C_{17}=15000$, $C_{21}=C_{22}=\ldots\ldots=C_{27}=10000$,
$P_{11}=P_{12}=\ldots\ldots=P_{17}=0.1$, $P_{21}=P_{22}=\ldots\ldots=P_{27}=0.2$;
$C_{18}=C_{19}=\ldots\ldots=C_{115}=8000$, $C_{28}=C_{29}=\ldots\ldots=C_{215}=4000$,
$p_{18}=p_{19}=\ldots\ldots=p_{115}=0.2$, $P_{28}=P_{29}=\ldots\ldots=P_{215}=0.3$;
$C_{116}=C_{117}=\ldots\ldots=C_{139}=2000$, $C_{216}=C_{217}=\ldots\ldots=C_{239}=500$,
$P_{116}=P_{117}=\ldots\ldots=P_{139}=0.4$, $P_{216}=P_{217}=\ldots\ldots=P_{239}=0.5$;
$C_{140}=C_{141}=\ldots\ldots=C_{151}=400$, $C_{240}=C_{241}=\ldots\ldots=C_{251}=50$,
$P_{140}=P_{141}=\ldots\ldots=P_{151}=0.8$, $P_{240}=P_{241}=\ldots\ldots=P_{251}=0.6$.

According to Equation (1), we can compute:

$A_1 = C_{11} \times P_{11} + C_{21} \times P_{21} = 15000 * 0.1 + 10000 * 0.2 = 3500$
$A_2 = C_{12} \times P_{12} + C_{22} \times P_{22} = 8000 * 0.2 + 4000 * 0.3 = 2800$
$A_3 = C_{13} \times P_{13} + C_{23} \times P_{23} = 2000 * 0.4 + 500 * 0.5 = 1050$
$A_4 = C_{14} \times P_{14} + C_{24} \times P_{24} = 400 * 0.8 + 50 * 0.6 = 350$

According to Equation (2), we can compute:

$\sum_{i=1}^{7}(C_{1i} \times P_{1i} + C_{2i} \times P_{2i}) = 24500$
$\sum_{i=8}^{15}(C_{1i} \times P_{1i} + C_{2i} \times P_{2i}) = 22400$
$\sum_{i=16}^{39}(C_{1i} \times P_{1i} + C_{2i} \times P_{2i}) = 25200$
$\sum_{i=40}^{51}(C_{1i} \times P_{1i} + C_{2i} \times P_{2i}) = 4200$
$\rho_1 = \frac{24500}{76300} = 0.3211$, $\rho_2 = \frac{22400}{76300} = 0.2936$
$\rho_3 = \frac{25200}{76300} = 0.3303$, $\rho_4 = \frac{4200}{76300} = 0.0550$

According to Equation (3), we can compute:

$H = -(0.3211 * \log_2 0.3211 + 0.2936 * \log_2 0.2936 + 0.3303 * \log_2 0.3303 + 0.0550 * \log_2 0.0550) = 1.8034$

According to Equation (4), we can get $H_{max} = \log_2 51 = 5.6724$, and $H^* = H / H_{max} = 0.3179$

According to Equation (6), we can compute the result

$R = 1 - H^* = 1 - 0.3179 = 0.6821$.

When a very critical requirement change occurs, it will risk to the whole project. So project manager could set a Risk Threshold α according to his/her work experience. When $R \geqslant \alpha$, they should pay close attention to risk and take appropriate measures. The model needs the effect of missing requirement change and additional requirement change could obtain from 3 areas:

- Degree of development progress lags.
- Degree of product quality decline.
- Degree of Direct losses of funds

The probability could estimate from 2 aspects:

- User's actual situation.
- Project manager's practical experience

These data are not difficult to obtain, so the model is practical.

VI. CONCLUSIONS

Clients would inevitably propose request of software requirement change in the process of software development. Frequent requirement changes are fatal risk to the process of software development. Every very critical requirement change will give the whole project a significant impact. In order to mitigate risks and reduce the risk of harm, project manager should clearly define the scope of requirement in the start-up phase, strictly control the scope of requirement in the implementation phase and sum up the scope of requirement in the final stage, finally draw lessons from the project. The purpose of controlling requirement change is not intended to control the occurrence of change, but manage change to better deal with them, so that we can guarantee the success of the project in an orderly manner. The more uniform the degree of software requirements effect project, the smaller the impact parts of requirement effect the whole project, the risk is low at this time. Otherwise once very critical requirement change will result in large fluctuations of system. The risk is high at this time. Information entropy could reflect the uncertainty of system and uniformity of subsystem, so we can measure the risk of software project requirements change. This paper presents a entropy-based quantitative model to measure risk.

ACKNOWLEDGMENT

This paper is funded by the Open Foundation of Key Laboratory of Software Engineering of Yunnan Province under Grant No. 2011SE13.

REFERENCES

[1] J. Cao, L.M. Liu, Y. Wang, "The application study of function point analysis method for software requirement management," Project management technique, vol.9, Sep. 2007, pp.23-26.

[2] X.Guo,"Pre-sale project management of software system integration,"unpublished Master Degree of Engineering dissertation, Central China University of Sciences and Technology, Wuhan, China, 2004

[3] H.D.Wen "Requirements analysis and system design of comprehensive transmission networks management system," unpublished Master Degree of Engineering dissertation, East China Normal University, Shanghai, China, 2006

[4] J.Rong, H.Z.Liao, X.M.Zhang, L.H,Chen, S.Li."Measure developer turnover risk of software project using information entropy," Computer Engineering and Applications, vol.23, Aug.2009, pp.208-210.

[5] Ihara Shunsuke. "Information theory for continuous systems, "Singapore:World Scientific,vol.1,Oct.1993.pp.79-83

[6] Shannon,Claude E,"A Mathematical Theory of Communication,"Bell System Technical Journal,Vol.27, n. 3, May.1948, pp.379–380.

[7] SAMSON.D.E, "Automated assistance for software requirements definition," unpublished bachelor of science degree dissertation, George Mason University, Washington D.C, USA, 1989.

[8] ROBINSON.M.J, "Risk assessment in software requirements engineering: an event driven framework, " unpublished bachelor of science degree dissertation, George Mason University Washington D.C, USA,1995.

[9] ARMOUR.F.J.A, "cluster analysis and prototyping approach for the risk management of software requirements, "unpublished bachelor of science degree dissertation George Mason University, Washington D.C, USA, 1993

[10] MYERS.M.E.A, "knowledge- based system for managing software requirements volatility," unpublished PhD. Dissertation,George Mason University, Washington D.C, USA, 1988.

[11] Romono J J. A, "Test-based Risk Identification Method for Requirements Assessment," unpublished bachelor of science degree dissertation, George Mason University, Washington D.C,USA,1997.

[12] Donald M Y."An Early Indicator to Predict Requirements Volatility,"unpublished dissertation,George Mason University, Washington D.C, USA, 2001.