# Risk Mitigation And Management Scheme Based On Risk Priority

Basit Shahzad[1]Sara Afzal Safvi[2]

GJCST Classification
K.6.1. H.4.1. K.6.M

***Abstract*- Much effort has been put in order to identify the possible risks hindering the successful completion of software projects. Techniques in risk mitigation, management and monitoring plan devise the estimation process of risk likelihood and their possible impact on the progress of software project. Risk Mitigation, Monitoring, Management is a thorough and continuous process, which aims to bring the potentially losing project to the safer shore. Hence every phase of this plan is of equal importance. Generally more focus is maintained in the initial phases i.e. the identification and assessment of possible risks. Whereas formalizing a concrete avoidance / mitigation plan must also be devised, to ensure that risk do not mature in problem. A response should be ready in advance. Generally it is easier to indentify and assess the risk but to suggest suitable mitigation / contingency plan is far more difficult task.**

**The measurement of effectiveness of these mitigation / contingency plans should be well carried. It must ensure that after the execution of such plans the risk exposure is reduced or preferably eliminated. This can be referred as the feasibility of the mitigation / contingency plan, which is critically analyzed and measured for its effectiveness. This paper focuses on the prioritization and then handling and proposing the mitigation strategy for each risk factor**

*Keywords*-Risk mitigation, Risk Priority, Risk Management, Risk Handling, Software Risk management

## I. Introduction and Risk Categorization

Risk can either be avoidable or unavoidable. Hence risks can be categorized into two main classes. Based upon the priority level of any risk we can also judge if this risk is avoidable or not. So we can either build up mitigation or avoidance plan for that specific risk. [Table 1 shows the most prominent risks which may hinder the successful project completion].

Any Risk may be same in type but it may differ in different kind of software projects. For example Low estimation of cost may result differently in an embedded system software project where as it may behave differently in an Information System software project. It would be more convenient to calculate the cost of an embedded system. The reason is that embedded system would be having limited environment of functionality and narrow area of integration. Whereas as compared to this an information system would be involving lot of user types, broader integration area and vast environment (e.g. distributed enterprise systems)

| Risk | Probability | Part-Impact | | | | | Avg-Impact | Effected factors | Impact | Impact's probability | Overall Impact |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A | D | C | T | M | | | | | |
| 1. Requirements are not properly stated | 50% | 1 | 2 | 3 | 4 | 5 | 3 | 2,3,7 | 45 | 22.5 | 55.2 |
| 2. Low estimation of cost | 50% | 3 | 3 | 3 | 3 | 3 | 3 | 8,9,10, 11,12, 17 | 45 | 22.5 | 103.9 |
| 3. More stress of users than expected | 30% | 0 | 0 | 0 | 2 | 4 | 2 | | 12 | 3.6 | 3.6 |
| 4. Less reuse than expected | 30% | 0 | 0 | 3 | 0 | 0 | 1 | 2,17 | 3 | 0.9 | 57.4 |
| 5. Delivery deadline tightened | 30% | 4 | 4 | 5 | 4 | 4 | 4 | 7,10,11,15 | 84 | 28 | 136 |
| 6. Funding will be lost | 10% | 3 | 3 | 3 | 3 | 3 | 3 | 10,11 | 45 | 45 | 41.7 |
| 7. Technology does not meet expectations | 30% | 1 | 1 | 5 | 3 | 1 | 2 | | 22 | 6.6 | 6.6 |
| 8. Lack of training on tools | 10% | 2 | 2 | 2 | 2 | 2 | 2 | 4,9,12 | 20 | 2 | 11.2 |
| 9. Staff inexperience | 10% | 2 | 2 | 2 | 3 | 2 | 2 | 1,2,4,6,7,8,13, 15,17, 18 | 22 | 2.2 | 145.6 |
| 10. Staff turnover | 30% | 3 | 3 | 5 | 4 | 4 | 4 | 6, 17 | 76 | 22.8 | 61.3 |
| 11. Manager changes circumstances | 40% | 3 | 3 | 2 | 2 | 2 | 3 | 2,17,18 | 36 | 14.4 | 92.5 |
| 12. Backup not taken | 20% | 2 | 2 | 4 | 1 | 1 | 3 | 13 | 30 | 6 | 15 |
| 13. Actual data/document loss | 20% | 3 | 3 | 3 | 3 | 3 | 3 | 2,16,17 | 45 | 9 | 69.7 |
| 14. Flood, fire and building losses | 10% | 2 | 2 | 2 | 2 | 2 | 2 | 2,17 | 20 | 2 | 58.5 |
| 15. Too many development errors | 50% | 0 | 0 | 0 | 5 | 5 | 4 | 6,10,11,17 | 40 | 20 | 95.7 |
| 16. Developer run away with code/doc | 10% | 0 | 0 | 5 | 4 | 4 | 3 | 2,6,17 | 42 | 4.2 | 65.2 |
| 17. Low estimation of time | 50% | 4 | 4 | 4 | 5 | 4 | 4 | 1,2,6 | 68 | 34 | 83.5 |
| 18. Lack of intuition | 30% | 4 | 4 | 3 | 4 | 3 | 4 | 1,2,4, 15 | 72 | 21.6 | 69.5 |

**Table 1 Risk probability and over all impact [7]**

Therefore proposed mitigation and management plan would differ based upon project type. In this paper either mitigation or management plan is proposed for each type of risk based upon the fact of avoidance and un-avoidance.

*About*-Comsats Institute Of Information Technology*
*(e-mail[1]-Basit_Shahzad@Comsats.Edu.Pk)*
*(e-mail-[2]Sarah_Safvi@Hotmail.Com)*

### A. *Requirements are not properly stated*

Yet a major risk factor affecting project schedule, budget and quality is the ability to successfully elicit requirements and execute on them.

This risk is avoidable and can be mitigated right from the beginning if tight grip is maintained in requirement elicitation phase.

## II. Mitigation Strategy

1. Maintain clear understanding of stakeholder needs and their relative prioritization.
2. Bridge the communication gap between customer so they can not claim about their requests being misunderstood resulting in rework of demands to be implemented.
3. Divide the users into specific groups based upon organizational hierarchy and target each group separately. As each group would be having different requirement based upon their set of duties. For example managers would be interested in decision support tools where as operational work force would be interested in convenient transaction processing.
4. Use every possible method to understand what user says and what the analyst comprehends out of that, so that the requirement shall be validated by the end user.
5. Prototype Demos and screen shot can be showed to the user to avoid ambiguities. Confirmed user requirements shall be document and signed by all stake holders.
6. All stake holders specially customer must be told very clearly about the feasibility of particular requirement.
7. Joint Application Development (JAD) is a group based requirement elicitation and design technique. JAD mainly features an intensive structured workshop. Expected end users, Analysts, Developers and Projects managers attend the workshop. The workshop is headed by an experienced leader. The leader conducts meetings with managers and end users to clearly define the domain, scope and objectives of the project. This leader also determines participants of a JAD workshop. The output of this workshop is a document which contains the clear user interests determined during JAD session. [12]
8. Facilitated Application Specification Techniques (FAST) aim to further decrease ambiguities in requirement elicitation process. As developers and customer work as team member rather than behaving as opponent parties. This attitude generally ends up in confrontation and confusions which hinder the clear elicitation of user requirements. FAST brings a facilitator between the customer and developer who conducts a meeting and behaves as a mediator.

"During FAST meetings the following activities take place.

a) Product need and justification
b) Lists discussed and combined
c) Lists refined
d) Mini-specifications prepared for each list entry
e) Mini-specifications reviewed by all
f) Validation criteria for the product/system" [10]

### A. *Low estimation of cost*

Accurate cost estimation is still a bottleneck in software planning process. Several methods already exist for this purpose. Mainly there are two categories of models to estimate the cost of software projects i.e. Algorithmic and non-algorithmic. Moreover most of the models are based upon the size of software project to calculate the cost. Each model has its own strengths and weakness. Selection of model revolves around the accuracy of its estimates. Unfortunately the accuracy of these models is not satisfactory. Moreover accurate cost estimation is the biggest success factor as well as risk in software development cycle. Software cost estimation focuses upon three main dimensions i.e. Human Effort, Time Duration and monetary resources required.

Keeping the unsatisfactory level of accuracy of existing cost estimating models, one must carefully decide which software cost estimation model to use. Which software size measurement to use (lines of code (LOC), function points (FP), or feature point). A good estimate must be determined keeping the project characteristics in view.

### B. *Mitigation Strategy*

I. This risk can be mitigated by early selection of best available cost estimation model relevant to the project characteristics. *Empirical Cost Estimation model* utilizes the historical data about past projects. Therefore it shall be evaluated to confirm if sufficient historical data about the same kind of projects (similar processes, similar technologies, similar environments, similar people and similar requirements) is available or not.

II. Expert judgment shall not be relied too much upon, as poor measurement of project size may result in unrealistic cost estimation.

III. *Analytical model* is another alternative, which is based upon the rate at which developer solve problems and the number of problems available. Line Of Code or Function Points are analyzed for project size measurement. Realistically it is very hard to calculate the actual number of code lines prior to the completion of project. Therefore if inaccurate size is input then, obviously inaccurate cost estimate would be resulted.[2]

IV. The relationship between cost and system size is not linear. Cost tends to increase exponentially with size. The expert judgment method is appropriate only when the sizes of the current project and past projects are similar.

V. Size of the project shall be measured keeping the detailed Work Breakdown Structure. So that cost estimate may encompass every area of cost and effort.

VI. Whichever cost estimation model is selected, it must be approved by all the stakeholders. Moreover the inaccurate factor shall be clearly explained to all stakeholders so that in future any drift from the planned estimated can be justified without contention.

VII. Experienced cost estimators shall be appointed to avoid any errors during process of cost estimation by inexperienced estimators.

VIII. Accurate measurement of project size is very essential as it leads to the accurate cost estimation of the project. [2]

### C. *Contingency Plan*

Once the risk has matured into problem contingency plan can be executed for recovery.

Best approach can be to execute another iteration of cost estimation for overrun project. As proposed in [3].

I. If planned duration is overrun, then compression techniques can be followed by squeezing the activities on to the Critical path. [11]

II. The manager must analyze the reason of delay, if it might have been caused by inefficiency of the workers. The penalty shall be placed on them in terms of over time with out extra payment, therefore saving any further monetary cost overrun.

## III. More stress of users than expected

This type of risk is surely avoidable and has the least overall impact ratio as depicted in table 1.

User sometimes may behave differently and hence may produce difficulties. This may result in more pressure on the development of the project.

### A. *Mitigation Strategy*

I. If ample level of understanding has been developed with the user, then user must not stress upon unrealistic demands. Once requirements are clearly documented, the user may not find any capacity to further argue about the requirements unnecessarily.

II. Non functional Prototype can be shown to the user, if user becomes hyper to see the progress of the project.

## IV. Less reuse than expected

Initially the cost of project might have been calculated by considering a good ratio of software reuse. And eventually it may not be practiced in reality which emerges into a severe risk factor.

This is an avoidable risk therefore can be planned to mitigate as earliest as possible.

### A. *Mitigation Strategy*

I. The source of reusable software shall be determined before actually using the software. Following factors must be carefully analyzed first to see :

a) If re useable software is available in house.

b) If any contract is made with third party to provide re useable software.

c) If any cost is incurred for re using the software, shall be carefully estimated.

II. Once the source is well specified, it can be assured that software / re useable components would be well in time available. Specially when they are not available in house.

III. Any delayed provision of such components shall be compensated by the third party, which is in contract.

## V. Delivery deadline tightened

This risk has the second highest risk impact (i.e. 136) as depicted in table 1.

This may mature in delayed project delivery, therefore must be handled in time. Hence it may be avoidable, but once mature the contingency plan is proposed.

### A. *Mitigation Strategy*

I. Close project monitoring shall be implemented continuously throughout each phase of the project.

II. Even if project is over running a single day, workers shall work over time to recover in time.

III. Project manager must make sure that software process is followed strictly.

IV. Moreover the entire organization must have matured to senior levels of CMMI or ISO whichever quality assurance process is implemented.

V. It is evident that organizations who have well achieved maturity level can better avoid risk at the initial stages.

VI. Proper tools and methods of configuration management shall be well in practice so that any requirement change may well incorporated and may not result in the delay or schedule tightening.

### B. *Contingency Plan*

Tightened schedule would definitely result into pressure. As gone time can not be reversed. Therefore time loss can not be recovered rather extra burden falls on to the shoulders of the workers.

Although an iteration to revise the schedule can be made to increase the number of working hours per day and completing the work in restricted time slot. [3].

## VI. Funding will be lost

Before taking off the fuel tank must be assured for fullness. Non availability of the funding can result in catastrophic results, similar to the crash of flight. Therefore ample funding should be guaranteed.

Although this risk has 10% probability factor but if turns into reality then it may earn total bad name and irrecoverable project failures.

### A. *Mitigation Strategy*

I. The sources of project funding must be determined and agreed upon by all the stake holders in the very initial stages of the project life cycle. Rather at the feasibility study stage.

II. If project is financed by some bank loan then all the necessary terms and conditions should be in place and well documents.

III. It is better to have some insurance plan for contingency effort.

## VII. Technology Does Not Meet Expectations

This may effect the re usability factor as well. Some reusable components, which best fit the user requirement, are not adapted for technological incompatibility issues.
There can be lots of issues in this regard:

I. Insufficient skilled human resource for that specific technology.
II. More funding is required.
III. Difficult maintainability.
IV. Incompatibility with other components.
V. Evolution is not possible.
VI. Customer resistance for the technology.

### *D. Mitigation Strategy*

I. Selection of technology is done at the very beginning therefore all above mentioned issues must be addressed.
II. A checklist should be made and a thorough comparison should be carried out to determine the best suited technology.
III. All stake holders must be taken in confidence for the use of specific technology.

## VIII. Lack of training on tools

Lack of training can be compensated by different strategies which may avoid this risk. This is minor risk as depicted at the second lowest number in risk priority table. But surely may not be underestimated to carry its impact to next stages, which may eventually result in delay due to in experienced workers.

### *A. Mitigation Strategy*

I. If funding is low for the project, then project manager may compromise over less experienced staff, but it must be supplemented by in house staff training prior to the work starts.
II. Activity slacks can be utilized for the training of next task.

## IX. Staff Inexperience

This may also prove deadly for the project success. As at any stage delay can be caused by mishandling of tasks by inexperienced workers.

### *A. Mitigation Strategy*

I. Team members selection shall be done very carefully selecting only those workers who have good experience on the tools.
II. Experienced staff should be allocated to critical task which may ensure that no delay is expected and hence ensure the smooth and efficient completion of the project.

## X. Staff turnover

This risk may be rooted very deep in the psychology of the workers. Many of the factors including internal and external to the organization can affect the throughput of the workers. Though external factors can not be fully controlled but at least internal factors can be eliminated or either minimized.

### *A. Mitigation Strategy*

I. Workers should have strong motivation for work in terms of monetary or other rewards.
II. Workers should be appreciated for what ever effort they put in the project.
III. In case workers deliberately ignore their responsibilities, some kind of penalty shall be placed on them. Therefore there shall be some balancing threat to make them work.
IV. Close monitoring shall be kept to know the status of work done by each individual and obtain any kind of possible hurdles (e.g. sick leave, resignation plan).

## XI. Manager changes circumstances

Rescheduling may bring many disturbance for the workers as their personal life may also be effected. An other cause may be forcefully switch over to different tool may upset the worker and may shatter the confidence level. All these factors can be avoided.

### *A. Mitigation Strategy*

I. A software house should well maintain its team of skilled workers (i.e. Analysts / Programmers / testers).
II. Different programmer teams can be built for specific tools. Therefore a foot ball player shall not be forced to play tennis.
III. Managers must adapt allocation of tasks to such individual who are extremely confidant to carry out that task.
IV. A substitute worker shall always be spared so that in case of extra burden of over time can evenly be distributed among them.
V. Incase there are many work places scattered over the globe or nation wide, worker should be sent to the place of his/her desire. Parting from family or social circle may also disturb the worker emotionally and mentally.
VI. If project manager feels any discomfort in any of the worker, he/she must adapt an empathetic attitude towards him/her. And must try to find out the real root cause and may try to resolve the problem if possible.

## XII. Backup not taken

In an information and technology based organization, it is next to a folly not to take regular back ups for the precious data resources.
Back up is not only necessary for data recovery, but Project management and configuration management also rely on the data about all stages.

### *A. Mitigation Strategy*

I. Back up should be taken on regular basis.
II. Some authority shall confirm that backups are taken regularly and intermediate versions of data are not ignored or lost.
III. More over backups can be kept at multiple places. For this multiple back up servers can be employed at different geographical locations.

## XIII. Actual data/document loss

Although it is an irrecoverable loss, yet can surely be avoided.

### *A. Mitigation Strategy*

I. Back up shall be kept not only in the office building but at some other place as well so that in case of any natural disaster, it shall be recovered.
II. Data and software library are assets of the organization therefore shall be valued and accordingly legal documentations shall be maintained in case are stolen or deliberately damaged.

## XIV. Flood, fire and building losses

Natural disasters can not either be avoided nor informed before. Therefore any lost caused by such threats must be born and there shall some concrete contingency planning for them.

### *E. Contingency Plan*

I. Company assets must be insured to retrieve the loss.

## XV. Too many development errors

Development errors are natural to occur, but frequency should not exceed from a reasonable rate. This risk may be avoided successfully but detection of errors is not an easy task until the software is put through the testing phase.
Therefore this may come to the surface at the later stage i.e. testing phase.
Hence a mitigation as well as contingency plan can be devised for this risk.

### *A. Mitigation Strategy*

I. Employment of experienced programmers can prevent too many errors in the code modules.
II. Moreover an experienced programmer can produce a better piece of code, which can be more efficient in logic implementation and reusability.
III. Lots of errors can be detected earlier, therefore code must be tested concurrently for such types of errors.

### *B. Contingency Plan*

I. Errors can result in worst loss if detected after deployment of the system at the user end, as cost to fix errors after deployment is too high. Hence all possible errors should be tested and verified carefully prior to system delivery.
II. If error ratio is too large, the coders / testers may be put to over time to recover the errors.

## XVI. Developer run away with code/documents

This can surely be avoidable risk. And can be prevented following the below mentioned measures:

### *A. Mitigation Strategy*

I. Whenever some new employee is hired, a contract shall be signed clarifying the ownership of the code / design created by the employee.
II. There shall be some surety bond filled by the employee that he/she may not take away the technical material or shall not sell to other outside parties.
III. If so, there shall be some legal penalty to prevent such theft.
IV. Good configuration management shall be in place.

### *B. Contingency Plan*

I. Proper configuration management should be practiced so that if latest version is lost then at least one previous version remains available. So that project can be resumed from one step behind.
II. This may cause in little tightening of the schedule and therefore shall be prevented by the rescheduling of the work by putting over time effort.

## XVII. Low estimation of time

Likelihood of this risk is high as much as 50%, therefore the impact can effect the successful completion reasonably. Time estimation is as much complicated factor as cost estimation and faces many of the inaccuracies.
Hence the same sort of precautions shall be adapted as mentioned in Low Cost estimation risk section.

## XVIII. Lack of intuition

This factor may vary on individual basis. As some veteran project manager would be able to sense the likelihood of problem occurrence without any evidence yet emerged onto the surface.

### *A . Mitigation Strategy*

I. Inexperienced project manager should not be granted the steering of the ship. As captain of the ship must be strong nerved and must have a foresight to cope up with any problems hindering the smooth sailing.

## XIX. Conclusion

In this paper major focus is put to devise and suggest an effective response towards a risk so that it can be prevented rather than the need of cure. Much work has been done to assess the risks, but few relates to the development of accurate responses to the risks.
We have also investigated mitigation and contingency strategies considering the priority level of each risks and the

likely frequency and effect of each risk in any or all of the phases of software development life cycle.

This work is a contribution towards risk avoidance and a remedy measure is proposed for each type of risk. It has been observed that a large ratio of the software projects fail due to many risk factors. Those risk factors have been clearly identified and assessed many times. But still these risks mature into problems causing the project failure. We have proposed solutions and mitigation plans against each type of risk focusing at its specific priority in the risk listing and the probability. Mitigation and contingency plan may reasonably be affected by the likelihood and impact factor of each risk. Based upon these the risks have been prioritized [7]. Risks with smaller likelihood but greater impact or vice versa are equally important to be mitigated and controlled. Therefore risk priority can determine the importance of any mitigation or contingency plan to be activated. The timing of activation can also be determined by examining the risk priority.

This is essential to know that how many phases of the software development life cycle those plans should be spanning over. Risk may not be easily got rid off, it may decrease its likelihood in one phase and may eventually catch momentum in the other.

This paper may serve for the basis to further improve the risk mitigation and management strategies.

## XX. References

1) MITIGATING RISK WITH EFFECTIVE REQUIREMENTS ENGINEERINGA Borland White Paper
2) [Software Cost Estimationhareton Leung, Zhang Fandepartment Of Computing the Hong Kong Polytechnic University
3) Javed Iqbal, Basit Shahzad, "Iterative Project Scheduling – A Time Bound Technique", International Conference on Intelligent Systems & Networks (Feb 23-25, 2007).
4) [4] TOM ADDISON, "Controlling Software Project Risks – an impirical Study of Methods used by Experienced Project Managers".Proceedings of SAICSIT 2002.
5) Sommerville, I. *Software Engineering*, 7:th edition, Addison Wesley, 2004, pp 00-00.
6) CMMI Product Team, *Capability Maturity Model Integration (CMMI)*, Version 1.1, Staged representation, Software Engineering Institute, Technical report CMU/SEI-2002-TR-029, 2002.
7) Basit Shahzad and Javed Iqbal, Software Risk Management – Prioritization of frequently occurring Risk in Software Development Phases. Using Relative Impact Risk Model"
8) Roger S. Pressman "Software Engineering: A practitioner's approach" 5th Ed. McGraw-hill international.
9) Hewijin Christine Jiau**,** Chia Hung Kao**,** Kuo Chen Wu, Kuo-Feng Ssu
10) "An Analysis Model for Software Project Development"2006 IEEEKarim Dad Atif Neuman Jamil, "Enhanced Facilitated Application Specification Techniques (eFAST)" International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR'06).
11) BobHughes & Mike Cotterell, Software Project Management, 2nd Edition, McGraw-hill international
12) Yihwa Irene Liou, Minder Chen, "Integrating Group Support Systems, Joint Application Development, and Computer-Aided Software Engineering for Requirements Specification"1993 IEEE