

An Enhanced Risk Analysis Model (ERAM) of Riskit

Ajayi S. W.

Computer Science Department,
School of Computing &
Engineering Sciences
Babcock University, Nigeria

Adekunle Y. A.

Computer Science Department,
School of Computing &
Engineering Sciences
Babcock University, Nigeria

Akinsanya A. O.

Computer Science Department,
School of Computing &
Engineering Sciences
Babcock University, Nigeria

Eze M. O.

Computer Science Department,
School of Computing & Engineering Sciences
Babcock University, Nigeria

Awodele O.

Computer Science Department,
School of Computing & Engineering Sciences
Babcock University, Nigeria

ABSTRACT

An exploration of riskit analysis graph (RAG) as a major technique of Riskit method is presented in this work with the aim of enhancing its capability for better risk identification (and management) and subsequently, contribute to software delivery time. The study begin with a brief background of the riskit as a major tool in risk analysis; pointing to the need for an enhancement of the tool and associated benefits plus disadvantages. After this, a review of closely related works in the field of study is presented leading to identification of some perceived limitations and challenges in the generic Riskit methods (RAG inclusive).

Next, an analysis of a typical riskit analysis graph process vis-à-vis its main components is presented. Using the stepwise approach to risk profiling, a prototype of the intended model called the “enhanced risk analysis model-

ERAM”- is presented based on risk ontology and prognosis states. The ERAM was developed in phases through leaning on the basic approach of risk models which comprises of a generic four steps –establishing the likelihood of occurrence of risks in the task pool, identification of major variables for measuring the impact (should it occur); a computer simulation is performed leading to a well defined risk profile and finally, a conclusion was drawn on the fact that Riskit can actually be extended through a deep analysis of its process components.

General Terms

Risk, Software / IT projects, quality, delivery time

Keywords

Riskit methods, RAG, Enhanced Risk Analysis Model (ERAM), risk identification, RTAS , software quality

1. INTRODUCTION

The Riskit method developed in 1996 by Kontio H. Englund and Victor R. Basili at the Maryland University, United State of America is one of the known risk management methods [13]. The method is a comprehensive one which is based on strong theoretical principles. Over the years, the riskit method has proven to be dependable in the delivery of good result when used by the software development team. Amongst its notable functions as explained by [10], “is to support systematic and logical risk analysis by the use of a graphical formalism to support qualitative analysis of risk scenarios before quantification is attempted”. Again in Riskit, risk level is done based on the availability of historical data or risk estimate’s accuracy which gives ample supports to multiple goals and stakeholders.

Based on the work of [13] and [10] work on “Risk Knowledge Capture in the Riskit Method”, when the riskit is implemented in a software project, it helps to avoid many of the limitations and challenges that are common to other risk management methods in software development process.

The Riskit analysis graph (RAG) technique belongs to the Riskit method ([8]; [5]). The technique has been a dominant method for managing risk for years. As a matter of fact, in software engineering field, risk and software managers did not see any need for improvement because it is almost impossible to prove that the method is an imperfect one. It looks like an encompassing risk management method already. However, the aim in this study is not to prove otherwise, but to figure out ways of enhancing some of its limitations for better performance. Before proceeding on the task, some of the benefits and disadvantages of the technique are identified.

1.0.1 Merits of RAG

Where RAG process is used the different risks within the whole project is broken down into four major components namely: “factors, events, outcome of an event, reactions, and effects on the overall goal of the developmental process” [7] & [6].

The analysis of each of these components shall be presented later in this work.

Based on [13] and [7] amongst other basic points, it is good to note that the riskit analysis graph is a flexible method, rooted in strong theoretical basis and tends to avoid many challenges and limitations common to other risk management methods. Perhaps this particular point can be said to be a major strength of the Riskit. Aside software engineering, the Riskit can be applied in other fields such as business.

Secondly the fact that riskit operates basically on information supplied to it helps its capacity to prioritize risk with the aid of Pareto ranking technique before picking the most critical.

Thirdly, the method permits visual representation, along with formal and detail documentation of all risk areas which cumulates in clarity in better understanding and communication amongst all stakeholders.

Last but not the least is that it uses the idea of utility loss to categorize the loss related with risk.

1.0.2 Prominent weaknesses

Amongst others, Riskit is known to be weak in the following area.

The first one is, risk prioritization during initial risk evaluation is based on their (that is risks) **probability and loss**.

Riskit documentation process is a detailed one. However, the **cost** of keeping and maintaining such a detailed documentation is usually very high

1.1 Aim and Objectives

The aim of this work is to examine the Riskit method along with Risk analysis graph (RAG) with a special interest in enhancing its performance. In line with this, the following basic objectives would be considered.

1. Analyse the Riskit method and also present the link between Riskit method and RAG which is one of its techniques.
2. Evaluate the efficiency level of the risk analysis model.
3. Propose a design of an “Enhanced Risk Analysis Model(ERAM) of Riskit”

1.2 Problem Statement

As analysed earlier in [1], the concept of risk in software project is an abstract and subjective one which needs to be adequately communicated if the thought (or plan) to attain utmost quality for the intended product must be upheld. This is even more important as most times requirements for products being developed are vaguely understood, in some cases plans to address risks may not be solid from start and cost benefit analysis are rarely done; all of which may contribute to delayed delivery or absolute failure in the end.

Though, there are several risk management methods and techniques that have been developed in the past to help with risk and risk areas of software project process; however none of these methods is designed or guaranteed to surpass others during full implementation. They all have one limitation or the other. For instance, some methods could be used to predict risk from the start of a project while in some, there may be need to wait for risk to actually happen before trying to alleviate the effect.

Again most old generic models are saddled with very high overhead cost due to the fact that same number of steps will be used in large, medium and small development during implementation.

Loss of project data is a usual occurrence during software project, system or during platform change. Some of the major factors named as culprit in this regard are the lack of automation and technicalities. These factors sometimes also make implementation almost impossible. Method like the Riskit has been found to work based on information supplies making the risk prevention process slow.

To this end, this work aims at proposing an adaptable form of Riskit method (here called ERAM) for risk prognosis in software projects. This will hopefully help to speed up risk prediction process, delivery time and consequently contribute to product’s quality in the end.

1.3 Methodology

The center of this work is an in-depth analysis of RAG as a major technique in managing risk. In what follows, the stepwise methodology, tool and “hypothetical model” is presented so as to create a general idea of the process leading to the intended design.

Basically, the methodology used in developing this work is through:

1. Establishing a link between this present work and an earlier one done titled “Software development top models, risks control and effect on product quality” [1].
2. Then views (diagrammatically) and comparisons are presented of the generic riskit process and a technique of Riskit method known as risk analysis graph (RAG) as shown in the literature.
3. Fact finding and analysis of existing work done in this paradigm.
4. Next is the presentation of an analysis of a typical scenario where the database aspect of a software development wasn’t ready as scheduled, the event that arose from this, the reaction and the effect set.
5. Presentation of the proposed design of ERAM.

2. LITERATURE REVIEW

This section of the work presents a review of previous related works done on Riskit analysis method (including its evolution) and establishes the gaps in literature. It is also very significant to state that several works have been done in this area; however, for the purpose of this present study, the review is narrowed down to only the notable closely related ones.

2.1 The View of the Riskit Method

As stated earlier in the methodology, it is considered very necessary to present a view of the generic Riskit method (here identified as the circular RAG) as seen in [10] and that of [6].

There are two views of the Riskit. One is identified as the circular view (the riskit method) and the other is a technique (graphical presentation) of the Riskit method called the Riskit analysis graph (RAG). Aside these, other variations may exhume see figure 3 as a result of improvement on the generic model.

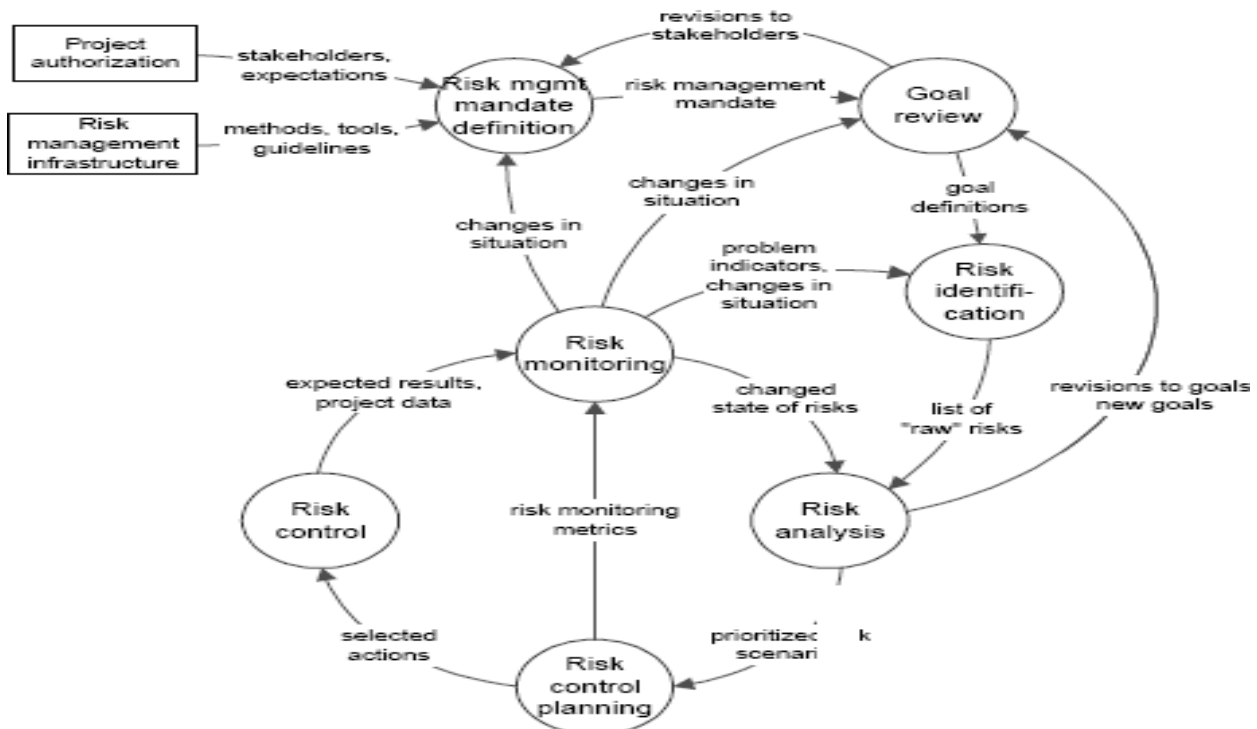


Figure 1: The Riskit method Source: [10]

2.2 EVALUATION OF RELATED WORKS

The very first to be considered is the work of [13] titled “an insight into riskit method” which began by defining risk management and outlining the various methods that have been used to analyze risk factors within projects (software projects inclusive) in the past. As stated by the authors, the aim of the work is “to give a brief description of most comprehensive risk management method known as RISKIT METHOD”.

In line with this aim, the authors were able to provide a concise analysis of the riskit method under eight distinct subheadings; starting with how risk is defined (including its management mandate definition), identified, monitored and controlled when riskit is implemented in a software project. Although, their analysis seems to have achieved what it set out for; however, it is expected that “strength, weakness, opportunity and threat – SWOT analysis” would be presented in the analysis; but this was not done; just few points were mentioned as advantages and disadvantages.

In [9] emphasis is laid on the strength of the Riskit method in combining sound principles into a consistent process and set of techniques and also provides clear description of risk in the Riskit method. It further stated that the Riskit method has been appraised in a number of experimental study and in several project or organizations during the past couple of years.

According to the author, though the findings from these studies are not yet conclusive, the feedback shows that the riskit method is practicable in real projects.

Based on [3] work on “Intelligent Risk Management Tools for Software Development”, software development presents many strategic opportunities but also plagued with equal level of uncertainties and since in most cases development and overhead costs are irrecoverable, it is therefore necessary to

device means of managing the uncertainties. The major focus of this work was to “demonstrate, through literature and current technology review, the need for intelligent and adaptive tools for risk management. Analysis of the framework of “Software Engineering Institute (SEI)’s Software Risk Management (SRM) Methodologies, Capability Maturity Model Integration (CMMI) and the Project Management Body of Knowledge (PMBOK)” and the two approaches to risk (and or project) management were presented. Though, having outlined the Riskit process and the elements present in it; the authors presented the need for intelligent risk management tools through the Neural Network approach. However, the work only presented the needs for intelligent tools by proposing two frame works namely: “neural networks and intelligent agent based”, no design or structural model was presented to aid understanding of their work. This needs to be improved on.

In [10] titled “Risk Knowledge Capture in the Riskit Method”, a blend of the Riskit risk management procedure and the experience factory was presented vis-à-vis the experiences gained from applying the approach in practice. The work was used to present a background and inspiration for risk knowledge acquisition and also enumerate the many useful characteristics of the Riskit method that can support this. Further analysis on how the Riskit method can be incorporated into Basili’s Experience Factory (EF) and Quality Improvement Paradigm (QIP) (see figure 3) was also done. Again, the authors were able to identify (through this work) that unforeseen problems account for the majority of challenges in software project and subsequently results in bad product quality and missing functionality amongst others. Perhaps this particular work is the closest literature to the proposed conceptual focus and main design in that it not only try to revise the authors’s initial and pioneering design of 1996 presented in both [10] and reviewed in [6] it also

provided a great attempt to improve RAG technique and consequently quality.

Next, in what follows, the main gaps observed from the reviews are presented. But first the diagram below (figure 2) shows how risk is defined in Riskit method.

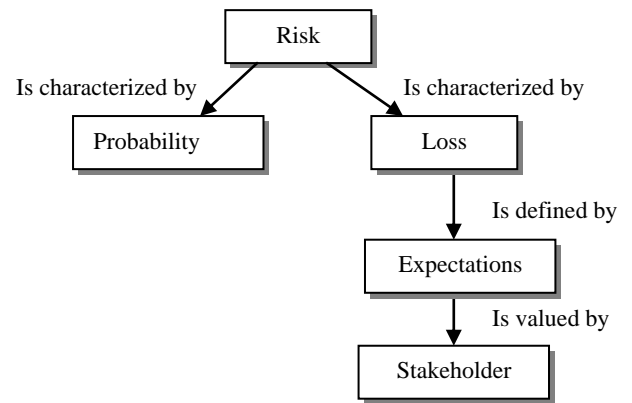


Figure 2: Showing Risk Definition By Riskit Method Source: [13]

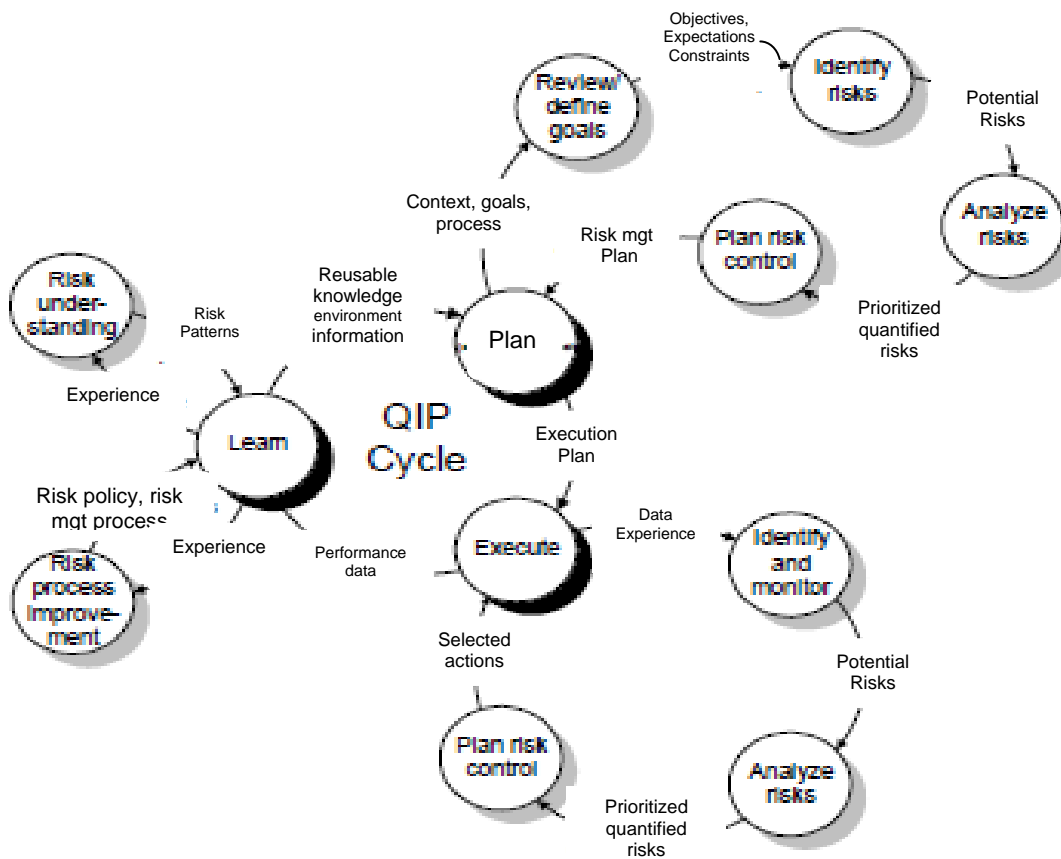


Figure 3: The mapping between Quality Improvement Paradigm (QIP) Cycle & the Riskit process Source: [10]

2.3 GAPS

Based on the literatures evaluated above and other sources consulted for this work, the following gaps in research were observed. These gaps will be leveraged on in order to achieve the listed objectives for this work.

Explicit analysis of [3] and that of [7] both portrayed the Riskit process (or methods) and its technique – RAG is a well-defined risk management process that works better based

on available and supplied information. This assertion is also supported by [9] explaining Riskit method as a “tool which combines sound principles into a consistent process and set of techniques and also provides clear description of risk”. However, the Riskit method though work with well define process and steps, automation of the processes involved still lack adequate attention.

[9] Further emphasized that Riskit methods and technique is now practicable in real life situations and projects. According to the authors, findings from these studies are not yet conclusive; decisions on if it can (or not) be implemented in real life were based on feedbacks from empirical studies. Hence, this also is worth re-examining.

Again, developing the graph takes a lot of time due to too many information, making the whole process lengthy. It is believed that this is due to less attention given to the automation of the general process. Hence in the proposed design, the proposed system helps with the risk identification by testing for sensitivity of tasks and the surrounding activities and then take proactive and spontaneous action to forestall occurrence of hazard along development process.

In the next section, an example of a typical analysis done using the rag process is presented.

2.4 Example Analysis Done Using The RAG Process

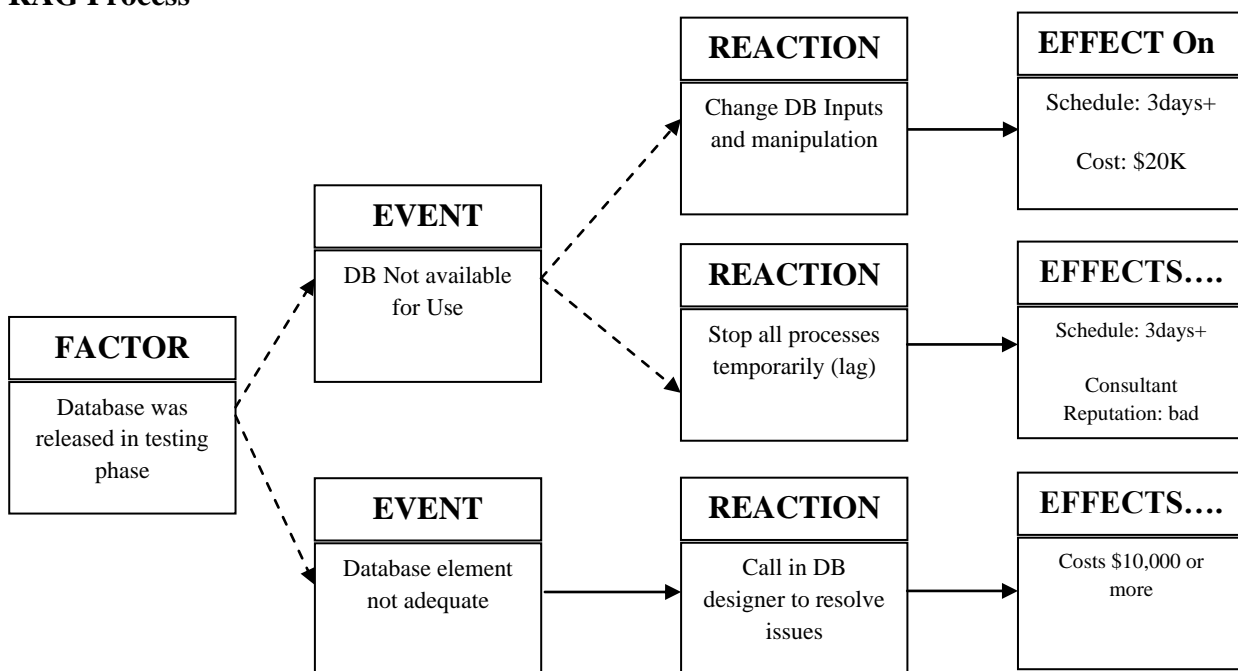


Figure 4: showing a typical implementation of the RAG process

2.5 The Generic Rag Components Explained

From the analysis of a typical scenario of the RAG process presented above, four major components were involved. These are: the factor, event, reaction and effect. These four components are the basis of risk analysis of the RAG process. Based on the perspective of [7] and [8] the analysis of each of the component is presented below.

a. Factor components

These are the characteristics that may affect the probability of a negative event happening. Examples of these are:

- Inexperience of team members or lack of adequate skills.
- Introduction of new methods and tools

The following is an example of a typical scenario where the database component of a software development process was not ready as scheduled, the event that arose from this, the reaction and the effect on output or product quality.

Risk Factor: Database was released at a wrong time/later than planned

Risk Event: Database not available for use or elements not adequate

***Outcome: system out of operation for about 4 hours

Risk reaction: change DB inputs and manipulation

Stop all processes temporarily

Call in DB designer to resolve issues

Effect: added cost, schedule extension.

The above process then yields figure 4.

- Constant change of requirements on the part of the user

b. Event components

This refers to a stochastic event that represents happenings of a negative or unpleasant impact. For instance

- Crashing of a main system
- Withdrawal of a key team member
- A major change in the requirement

c. Reaction components

These are possible action resulting from the risk event and (or) its outcome.

Examples are: System reconfigured after crash, database redesigned etc

d. Effects / Effect set components

This describes the impact of reaction. It is the effect of the occurred event. It could be in monetary terms or otherwise. For example: loss in functionality, added cost of \$20,000 etc

3. THE ENHANCED RISKIT ANALYSIS MODEL(ERAM)

The initial sketch of the enhanced riskit analysis model - ERAM is presented in figure 7. But before the explanation of flow of operation, an analysis of some terminologies used in the design are presented.

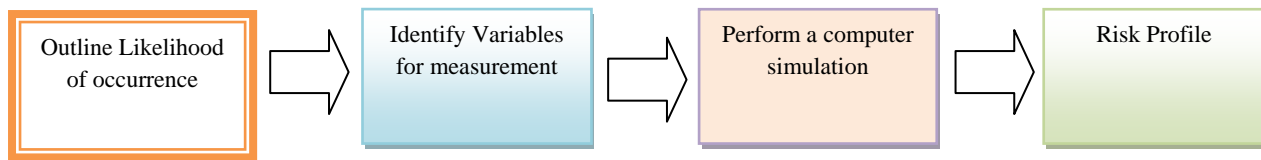


Figure 5: showing risk activities steps

The followings are some adaptive or adoptive words used to either define or explain some important concepts in the design and some other parts of the write up.

1. **Ontological:** This is adapted from ontology (as related to computer science) which means “**Taxonomy** for the partial ordering of C in generic concepts or narrower concepts (where C = a set of concepts). It also means formal naming, definition of a concept and their relationships or shared understanding of some domain” [2].

In this work, Ontological stage is where all software project challenges, risks and risk areas are identified, categorized and attached to a name. It is used to qualify and depict the left hand side (LHS) of the added part of the ERAM.

2. The “**RTAS**” is an abbreviation of “Risk Task Automation Stage” which is the general name given to the part of the ERAM where the different task and activities from the ontological stage are automated before the result is sent to the knowledge base.

The RTAS is comprised of two major sections – the main task automation and the knowledge base.

3.1 Analysis Of Design Steps

To arrive at the proposed model, first the basic approach of risk models (figure 5) as a stepwise approach is considered. According to the diagram, the first step is to list likelihood of occurrence, and then identify major variables for measuring the impact; next a computer simulation is performed and lastly, arrive at a clear and well defined risk profile [4]. Based on this understanding, in the ERAM design, the process is divided into two major parts.

- I. The original or generic Riskit process model, shown as a circular series of processes on the left hand side (LHS) of the model.

- II. The supplementary part is shown as a block diagram on the right hand side (RHS) of the model. This part is further divided into and named (for the very first time) the Risk “Ontological” stage and the RTAS (Risk task automation stage).

The ontological stage is the stage at which all tasks and activities are identified and named.

All resources allocated to and for the different tasks are also identified with the tasks or activities.

From the problem indicator stage in the generic Riskit method, the task is broken down into two (as indicated by the arrows in figure 7). As shown, all already identified risks or problem areas go into “B” where the fuzzy set is used to further classify it; while all tasks and activities planned to be carried out in the project goes into “A” where they will be analysed and separated as either tasks, or resources.

Note also that the categories of task that are considered in B include the following:

- a) Task that are presumed difficult.
- b) Tasks that take longer time to accomplish.
- c) Tasks that require more than one resource to complete (this is because if a task requires more than a resource, it is likely to delay another task waiting to use the same resource thereby creating a deadlock hence it will require a different algorithm to resolve).

The processing part begins with “C” where the automation of different tasks and activities is performed to determine sensitivity and proximity of tasks to danger. By doing this, determination of the critical path is also made easier. The critical path is the “longest sequence of tasks or activities in a software project plan, that must be finished on time (as scheduled) for the whole project to complete as planned” [11].



Figure 6: Design Sets

After the task automation in “C”, the result is sent to the knowledge base where (together with existing information), it is analysed then used as bases for decision making.

4. FINAL REMARKS

4.1 Contribution to knowledge

The RAG is undoubtedly a very good risk model. However, from research and the subsequent ERAM design (presented in figure 7), it has been shown that an aspect of Riskit and RAG

can be extended to act autonomously in helping with identification or determining and prognosis of risk (risky events, activities and all risk areas) without waiting to be fed information but by getting it directly from the constantly updated knowledge base created.

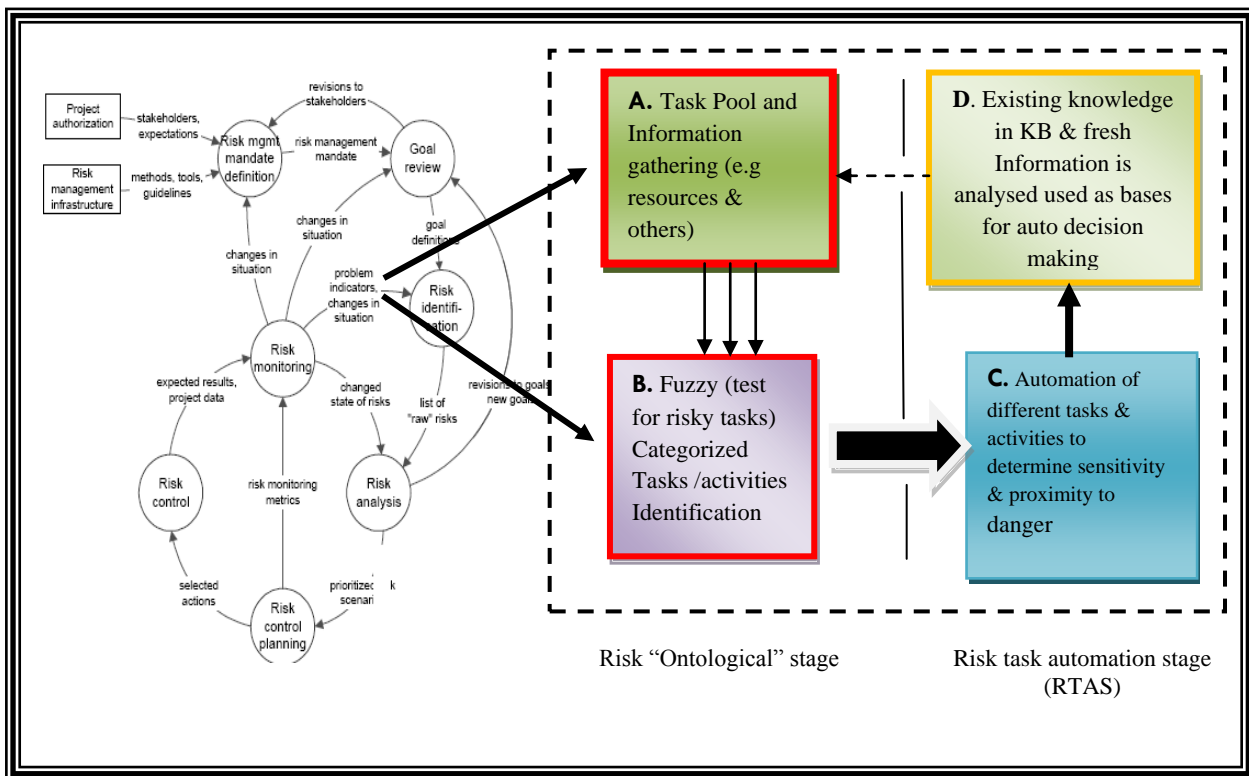


Figure 7: Proposed Enhanced Risk Analysis Model (ERAM)

The initial design reviewed the link between goal review and risk analysis to make room for the extension and improvement of the riskit method. A lot of works still need to be done beginning from this initial stage, but this is just an “eye opener” to realizing the bigger aim in the proposed model; which is to help improve the delivery time and the overall product quality. Hopefully, the proposed design will be improved in the nearest future to attain this goal.

4.2 Conclusion And Future Work

All risks in software project have “deadly capacity” to reduce (or mar) both software project life span and the quality of the intended product. Hence, any identified risks, risk events and activities or tasks with difficulty from the start of a project must be handled with care.

To this end, the ERAM is considered an excellent risk model that can serve in this regard.

In line with previous studies done in [1] and this present work, in the future, attention and consideration shall be given to the following.

1. Presentation of a comprehensive analytical view of the ERAM model shown here vis-à-vis the original risk identification procedure of the Riskit process.
2. An investigation will also be conducted into how feasible it is to use the fuzzy concept in the system to further help with the prognosis process for a near perfection (if not perfection) of the work since the Fuzzy set comes in before automation.

5. ACKNOWLEDGEMENTS

We acknowledge the excellent efforts and intellectual prowess of Kontio J. and Basili V. who originally invented the Riskit Methods (and its techniques) which serves as basis for this work. Many thanks to you.

6. REFERENCES

- [1] Ajayi W. , Adekunle Y.A., Awodele O., Akinsanya A. O., Eze M.O and Ebiesuwa seun (2017) : Software development top models, risks control and effect on product quality, *Global Journal of Computer Science and Technology*, Volume 17, Issue 3 version 1.0.
- [2] Busse J., Humm B., L'ubbert C., Moelter F., Reibold A., Rewald M., Schl'uter V., Seiler B.,Tegtmeier E., Zeh T.(2015) Actually, What Does "Ontology" Mean? A Term Coined by Philosophy in the Light of Different Scientific Disciplines, *Journal of Computing and Information Technology - CIT* 23, 2015, 1, 29–41
- [3] Dhlamini J., Nhamu I., and Kachepa A.(2009) Intelligent Risk Management Tools for Software Development, SACLA '09, 29 June - 1 July, ACM, Mpekweni Beach Resort, South Africa
- [4] Dumbravă V. and Jacob V.(2013) Using Probability – Impact Matrix in Analysis and Risk Assessment Projects, *Journal of Knowledge Management, Economics and Information Technology*, special issue
- [5] Edward H.(2008) Risk and opportunity of system change in organization (ROSCO) *presentations, University of Sunderland*, North East United Kingdom
- [6] Freimut B., Hartkopt S., Kaiser P., Kontio J and Kobitzsch W.(2001) An Industrial Case Study of Implementing Software Risk Management, **ESEC/FSE-9: Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering**, ACM , pp 277 – 287.
- [7] Hashemian V. (2003) *The Riskit Method For Software Risk Management*, University of Waterloo, USA
- [8] Kontio J. (1997) *The Riskit Method for Software Risk Management. Version 1.00*. CS-TR-3782. Computer Science Technical Reports. University of Maryland. College Park, Maryland. Available from <http://www.soberit.hut.fi/~jkontio/riskittr.pdf>
- [9] Kontio J. (1999) Risk Management in Software Development: A Technology Overview and the Riskit Method, JCSE '99, ACM, Los Angeles CA
- [10] Kontio J. and Basili V.R.(1996) Risk Knowledge Capture in the Riskit Method, viewed on 29/10/2017from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.5592&rep=rep1&type=pdf>
- [11] Office of Government Commerce- OGC (2012; 2013) Project in a Controlled environment (PRINCE 2), TSO, UK
- [12] Oxford Advanced Learners dictionary (2016). Oxford Advanced Learners, Oxford University Press, accessed on 27/10/2017<http://www.oxfordlearnersdictionaries.com/definition/english/>
- [13] Saif H. and Sadim M.(2015) An Insight into Riskit Method, *IJIRST –International Journal for Innovative Research in Science & Technology/* Volume 1 | Issue 12 | May 2015