

**Intelligent Decision Support (IDS) in Software Risk Management Based on Data Mining,  
Rough Sets and Decision Theory**  
*Mutheu, Rose and Wasike, Jotham*  
**Kirinyaga University, Kenya.**  
**Correspondence:** *rmunyao@kyu.ac.ke*

**Abstract**

*Risks are intrinsic to any project and risk-taking is a necessary component of any process of decision making. High risks from software projects threaten healthy development of any Nation because of the complex nature of projects. For sustainable development, we should focus on risk assessment and risk decision. Assessment of risks in most of the software projects has been done qualitatively ignoring the risk decision. Attempts at risk decisions have been based on individuals' rational opinions hence subjective. Previous reports have shown limited evidence on successful use of DSSs in practice. To address this anomaly, this study proposes intelligent decision support that provides more objective, repeatable, and observable decision – making support for software risk management. Software risk managers will be supported in gathering and analyzing evidence, identifying and diagnosing problems, proposing possible courses of action and evaluating such proposed actions. IDS is based on data mining, rough sets and decision theories which improve decision making in uncertain conditions. Risks will be looked at as identifiable and quantifiable possible events or factors from which negative or positive consequences may occur. The main sources of data will be a set of secondary data collected over time and knowledge of domain expert (s). The techniques used in this paper will provide efficient algorithms for finding hidden patterns in software risks and generate sets of decision rules to support decisions in software risks management.*

**Keywords:** Software Risks, Risk Decision-Making, Data Mining, Rough Set Theory, Decision Theory, Intelligent Decision Support.

**Introduction and Background**

A software risk management study by Kumar & Yadar, 2015 showed that industry-wide, only 16.2% of software projects are on time and budget, 52.7% delivered with reduced functionality and 31.1% cancelled before completion. The main reason for this large amount of less quality software and failure of software projects is the lack of proper software risk management and decision making. Software projects management entails keeping a balance between requirements, expectations, perceptions, opportunities and risks (Cunha, et al, 2016).

Software products are flexible and uncertain hence general approaches of project management are not appropriately applicable. Thus, "The success of software projects depends on how the managers deal with the problem and make decisions", Cunha et al 2016. According to the project management institute (PMI), poor decisions contribute to 47% of unsuccessful projects. Since decision making is a complex process which affects projects negatively if poorly done, there is dire need for researchers to establish support in decision making in project management. Such decision can be rational or otherwise, be based on explicit or tacit knowledge and beliefs and can yield optimal or less satisfactory solutions. Because software development lifecycle, generates of bulk streaming data which calls for application of big data analytics technologies in data collection and analysis, and many decisions have to be made concerning people, resources, processes, tools and techniques involved. These decisions can be less successful or otherwise subject to the various factors that affect the process.

Subjective analysis or expert judgment has been generally used in project risk management based on the experience of an expert which is not readily shared among different teams within an organization. It is thus critical to develop perfect modeling techniques that can provide more objective, repeatable, and observable decision – making support for risk management. This is achievable through use of an intelligent decision support system which should behave like a human consultant to support decision makers by gathering and analyzing evidence, identifying and diagnosing problems, proposing possible courses of action and evaluating proposed actions.

Decision making process is mainly geared at finding an optimal solution to a given problem by analyzing a finite set of alternatives, ranking these alternatives in terms of how attractive they are to the decision-maker and finding the best alternative for maximum expected utility. Since Artificial Intelligence has a greater computational information processing capacity and analytical approach, it plays a crucial role in extending humans' cognition when addressing complexity in organizational decision making. The aim of the Artificial Intelligence (AI) techniques embedded in an intelligent decision support system is to enable these tasks to be performed by a computer, while emulating human capabilities as closely as possible. Accuracy and consistency can be comparable to (or even exceed) that of human experts when

the decision parameters are well known. These techniques focus on enabling systems to respond to uncertainty in more flexible ways.

### **Problem Statement**

The outcome of any project is greatly affected by the decisions made at any stage of the project. Since decision making is a complex process, researchers should establish support in decision making in software development projects. Most Decisions are made through subjective methods hence they are unreliable. It would benefit more with an objective, repeatable, and observable decision – making support for risk management. In other cases, there is lack of clarity on how teams make and evaluate decisions from software inception to product delivery and refinement. The team of software developers who may be co-located or dispersed members adopt a discussion based approach thereby lacking a structured way of decision-making. In such teams there is rare usage of dedicated software tools for decision-making. The decisions made are thus affected by groupthink and divergence.

### **Literature Review**

This section seeks to summarize existing knowledge into coherent systems in order to give the new study direction and impetus. Any relevant theory to the area under study is considered. These theories could be from a single or multiple disciplines. This review is useful because it helps the researcher identify what is known about the research study and the type of knowledge available and helps establishing the theories that best guide the research work. This review determines if the research study is theory proven and how existing theories and findings are applicable in practice.

### **Risk Management in Software Projects**

Risks are uncertain meaning that they may or may not occur. If they occur, they cause undesired outcomes thus affecting the success of the project. This therefore calls for continuous management of the risks. Risk Management depends on the perception and recognition of sources of risk in all phases of a project. Managing risk in multiple project environments is important to enrich and qualify the information for project manager's decision-making (Wanderley, etal, 2015). Risk management comprises identifying, analyzing, planning and controlling events that threaten project environment so as to avoid or reduce the damage of these events should they occur. Risk Management, on its own does

not guarantee the success of projects, but increase the probability of project success by ensuring deadlines are adhered to, the project is inside the planned budget and the project goals are met. Software Risk Management is a cyclical and continuous process which involves the following phases-:

- a) **Risk Management Plan.** In this phase, decisions are made on how to plan the project's risk management activities, resource allocations, teamwork, and documentation standards.
- b) **Risk Identification.** In this phase, risks that might affect the project are determined and their characteristics documented. Many techniques to collect risk are employed.
- c) **Risk Analysis.** All project activities conditions are analyzed qualitatively to determine and prioritize their impact on project objectives. Quantitative Analysis is done involving determining the probability and risk sequences in order to estimate their impact on the project objectives.
- d) **Risk Response Plan.** Though Risks are always involved with loss, the possibility that the outcome of certain risks might be a gain is considered. This activity determines how to enhance opportunities and minimize loss.
- e) **Risk Monitoring.** This entails checking all risks identified and looking forward to new risks in the environment.
- d) **Risk Control.** The effectiveness of risk response plans is executed and evaluated. The success in this phase calls for well-defined schedules.
- e) **Risk Communication.** For successful accomplishment of risk management, there is need for communication.

## Bayesian Network

This paper will treat risks in software development projects as conditional risks since they depend on other factors like schedule slips, cost overruns, project scope and software quality. The probability of risks in software development project is therefore a conditional probability. The conditional probability of an event  $B$ , in relation to event  $A$ , is the probability that event  $B$  will occur given the knowledge that an event  $A$  has already occurred. According to Bayes rule,  $P(A/B) = \frac{P(B/A)P(B)}{P(A)}$

$P(A|B_1...B_n)$  has to be attached to each variable  $A$  with parents  $B_1, ..., B_n$ . Conditional probability will thus determine the probability of a software project failing given the evidence that there has been a risk (s) in the management of software project's time, cost, scope and

quality. For this to be reliable, the causal relationship between software projects failure and the causal factors is established using Bayesian network.

$P(\text{SW project failure} / \text{Schedule slips, Budget overruns, poor scope, poor quality})$  is calculated by applying

*Bayes' rule*, which states that:  $P(\text{SW project failure, Schedule slips, Budget overruns, poor scope, poor quality}) = P(\text{SW project failure} / \text{Schedule slips, Budget overruns, poor scope, poor quality}) * P(\text{Schedule slips, Budget overruns, poor scope, poor quality})$ . This equation will further be simplified as:  $P(\text{SW project failure, Schedule slips, Budget overruns, poor scope, poor quality}) = P(\text{SW project failure} / \text{Schedule slips, Budget overruns, poor scope, poor quality}) * P(\text{Schedule slips}) * P(\text{Budget overruns}) * P(\text{poor scope}) * P(\text{poor quality})$ .

Since the variables Schedule slips, Budget overruns, poor scope, poor quality are independent their prior probability is calculated from experiment and experience. The Bayesian Belief Network (BBN) is useful in representing and reasoning with uncertainties. The Bayesian networks (belief network, probabilistic network, causal network, and knowledge map) captures uncertain knowledge in a natural and efficient way in order to determine the causal factor that directly affects project outcomes by giving the correlations between risk factors and project outcomes. A BBN consists of two parts which include; 1) Qualitative part which represents the relationships among variables by the way of a directed acyclic graph, and, 2) Quantitative part which specifies the probability distributions associated with every node of the model.

BBN handles the situations where some data entries are missing or unavailable and can be used to model causal relationships. The update is easy whenever new knowledge is available. BBN are based on a set of conditional probabilities, and as evidence becomes available the expected probabilities of disruption occurrence are updated. The interrelationship among risk events is made explicit, allowing the user to trace the propagation of risks from one event to the next and ultimately across life-cycle phases. The BBN approach has proven to be a powerful tool when uncertainty is an important factor and allows easy visualization of the network through its graphical nature. This makes it easy to localize a problem or identify vulnerable areas in the network. Though powerful, BBN-based risk analysis is limited by its inability to handle feedback relationships. Due to linear

propagation of conditional relationships in BBNs, the reinforcing and balancing feedback relationships may not be fully captured.

## **Decision Theory**

Decision theory is a model designed to describe how individuals actually make decisions (behavioral decision theory) or how they should make decisions (normative decision theory). The normative decision theory begins with a set of mathematical axioms which formalize the logical and rational preferences. These Axioms classify accepted truths to ensure there is consistency and the chosen alternatives are independent of any other available alternatives. They capture the minimum requirements for rational decision making. Representation theorem allows the decision maker to quantify their preferences using utility function. The expected utility theory is prominent in making decisions under uncertainty. This is because such decisions based on expected utility theory are logical and self-consistent. However, they are not easily deployed in systems' engineering organization. Axioms are stated in terms of preferences in uncertain events. These axioms are stated below-:

- a) Completeness axiom; establishes that the decision maker can always decide between two alternatives otherwise the decision method suffer indeterminacy.
- b) Transitivity axiom; asserts that preferences must be linear. That is if  $A \succ B$  and  $B \succ C$  then  $A \succ C$
- c) Continuity axiom; implies that there are no infinite preferences
- d) Independence axiom; postulates that one alternative between two alternatives should not be impacted by availability of other alternatives

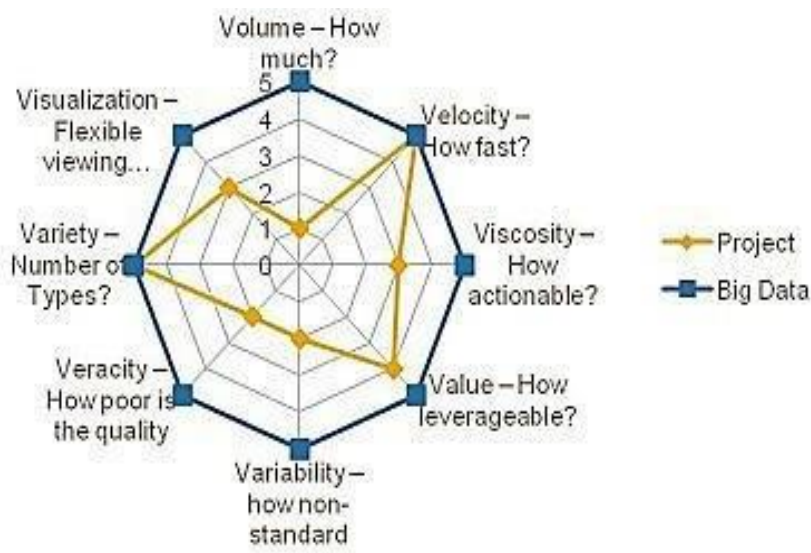
A rational decision maker should select the alternative with maximum expected utility where there are more than one options.

## **Big Data Analytics**

Big data deals with huge data which are unstructured. Using analytics tools, it can be chunked down and analyzed to provide valuable solutions. Big data analytics is widely used in all areas which deal with analyzing data especially unstructured data. Analysis of such data are difficult and can be implemented by use of different platforms and tools. Big data analytics has wide applications in software project risk management and can be used to predict the risk encountered in software project and provide objective recommendation, Usually Big data is referred to in terms of its V's characteristics which have evolved over

time. To date there are 7 V's characteristics of big data namely Volume, Velocity, Variety, Veracity, Variability, Value and Visualization (Rekha & Parvathi, 2015). The volume is the measurement of the amount of data generated in a day, velocity the speed with which the data is generated, and variety the type of data generated which can be structured or unstructured data though big data deals with unstructured data mostly. Veracity of data provides accurate data for processing both quality and understandability while variability of data deals with different data types. Visualizations is the representation of data in so many variables and parameters that makes the findings clear. Value of the data is how meaningful the data is and plays a big role in big data analytics because according to big data evangelists, if you can't put meaningful of data you can hardly put a monetary value on big data. Figure 1 below shows the 7 V's characteristics of Big Data

**Fig 1: 7 V's Characteristics of Big Data**



For Data mining to be highly effective, it should employ one or more of the data mining techniques which include the following

- i. **Tracking patterns;** This entails recognizing patterns in a data sets. Some deviations in the data which is happening at regular intervals or an instability of a certain variable over time are established.
- ii. **Classification;** This technique seeks to collect various attributes together into distinct categories, which you can be used to draw further conclusions, or serve some function.
- iii. **Association;** Association is related to tracking patterns, but is more specific to dependently linked variables. It entails looking for specific events or attributes that are highly correlated with another event or attribute.
- iv. **Outlier detection;** This helps to identify anomalies, or outliers in the data.
- v. **Clustering;** Clustering is similar to classification, but involves grouping chunks of data together based on extent of their similarities.
- vi. **Regression;** Regression is used as a form of planning and modeling to identify the likelihood of a certain variable, given the presence of other variables. Regression focuses on helping uncover the exact relationship between two (or more) variables in a given data set.
- vii. **Prediction;** Prediction is one of the most valuable data mining techniques. It's used to project the types of data you'll see in the future. In many cases, just recognizing and understanding historical trends is enough to chart a somewhat accurate prediction of what will happen in the future.

### **Research Methodology**

The study was exploratory based on the objectives of the study. According to Kothari (2004) Exploratory Study mainly seeks to formulate a problem for more precise investigation or for developing the working hypotheses from an operational point of view. The study will emphasize on the discovery of ideas and insights.

This study is applied research based on the application of the study, designed to solve practical problems of the modern world, rather than to acquire knowledge for knowledge's sake. The goal of the applied study is to improve the human condition Kothari (2004) Considering the type of Information sought, this is a Quantitative Research involving quantitative generation of data and analysis of the data.

This study was based on Secondary data whose sources included the following

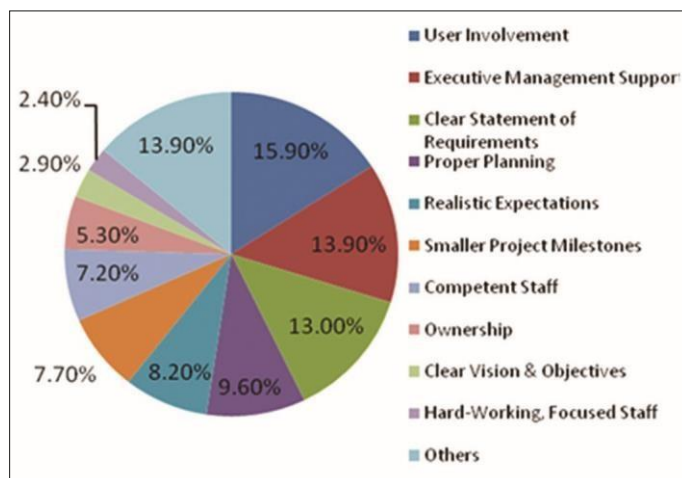
- i) Open source software companies and communities
- ii) Survey of related literature in scientific publication for software development
- iii) Experience survey of domain experts
- iv) Analysis of Case Studies in Large software development

### **Data Processing and Analysis**

Software projects fail when they do not meet the criteria for success. Most of the IT projects run over budget or are terminated prematurely and those that reach completion often fall far short of meeting user expectations and business performance goals (Kaur et al, 2011). The three major key factor of project success are delivered on time, on or under budget, the system works as needed. However, the ability of the system to meet the user requirements is the basis for the success of software projects since success or failure of the project depends on how the software requirements process was carried out. The cost or the risks involved in a poorly engineered requirements process are great and sometimes irreparable (Hussain et al , 2016)

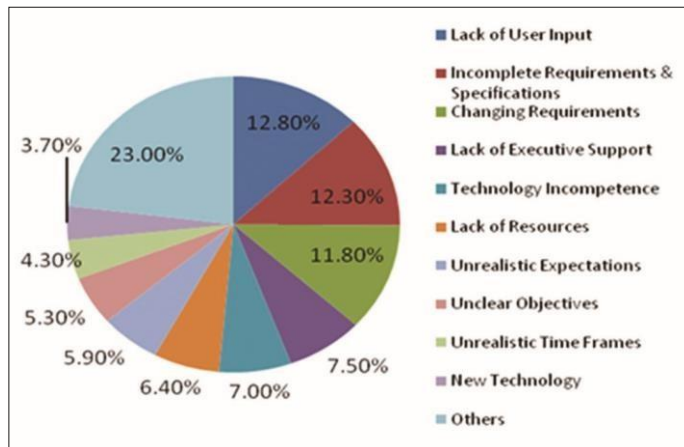
Statistics of why projects are impaired and ultimately cancelled shows that incomplete requirements and lack of user involvement contribute more (Project Smart, 2014) as illustrated in figures 2-4 below.

**Figure 2: Software Project Success Factors**



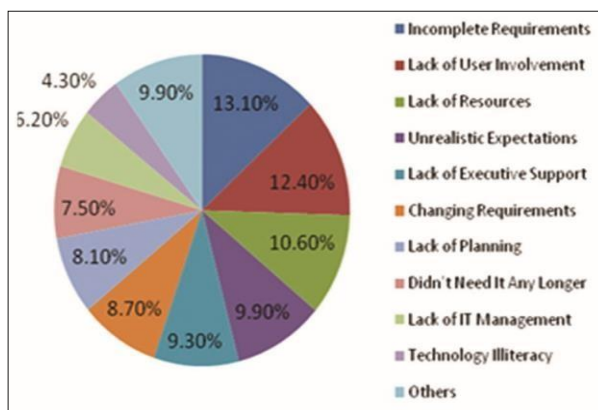
Source: Project Smart (2014)

**Figure 3: Software Project Challenge factors**



Source: Project Smart (2014)

**Figure 4: Software Project Failure Factors**



Source: Project Smart (2014)

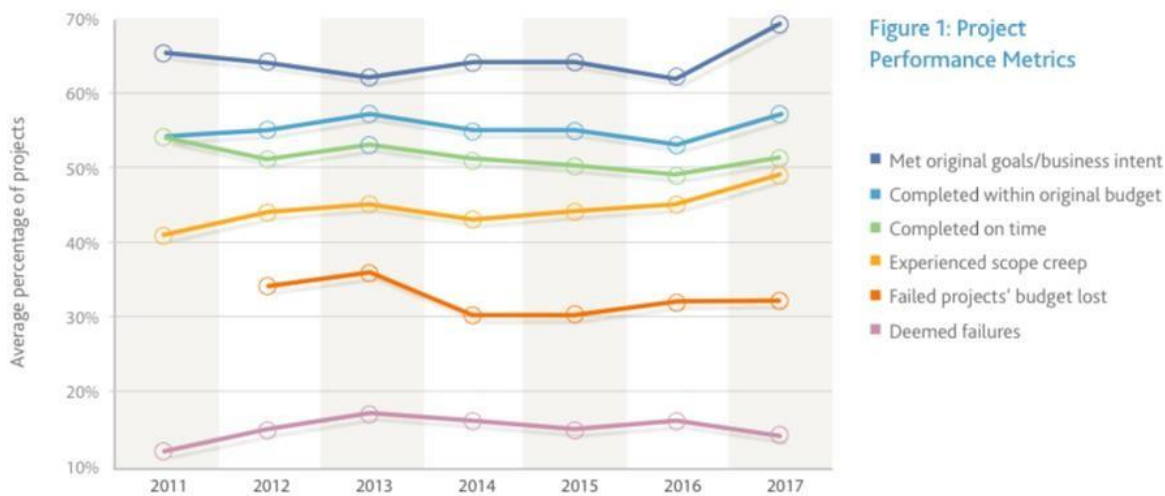
The 2018 Project Success Survey carried out by PwC Belgium ("Project Success Survey 2018 Driving project success in Belgium," 2018) shows that there are Twelve Elements of Project Success.

**Figure 5: ("Project Success Survey 2018 Driving Project Success in Belgium," 2018)**



The chart below shows the project performance metrics between 2011 and 2017 as per a survey carried out by the PMI in 2017.

**Figure 6: Project Performance Metrics.** Reprinted from *PMI's Pulse of the Profession 9th Global Project Management Survey*, by Project Management Institute, 2017



The CHAOS Report by The Standish Group, 2015 from the software development industry, confirm that a larger project is harder to complete successfully than a smaller project. This is because larger projects have more resources and broader scope hence increased.

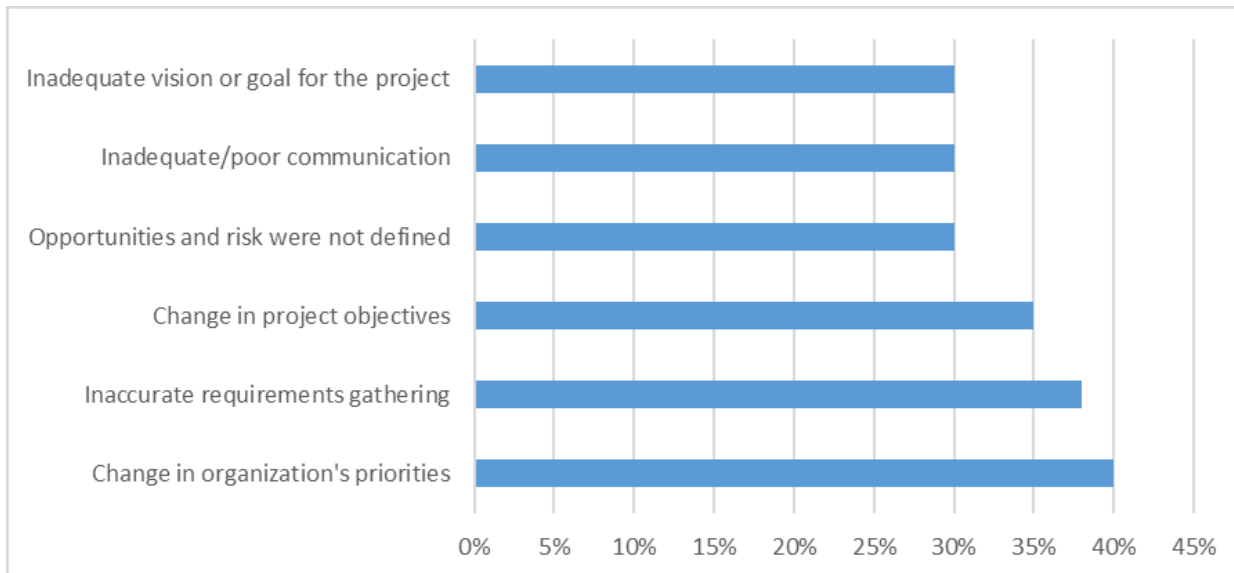
**Table 1: Agile Vs. Waterfall. Adapted from 2015 Chaos Report, By The Standish Group, 2015**

| SIZE                 | METHOD    | SUCCESSFUL | CHALLENGED | FAILED |
|----------------------|-----------|------------|------------|--------|
| All Size             | Agile     | 39%        | 52%        | 9%     |
|                      | Waterfall | 11%        | 60%        | 29%    |
| Large Size Projects  | Agile     | 18%        | 59%        | 23%    |
|                      | Waterfall | 3%         | 55%        | 42%    |
| Medium Size Projects | Agile     | 27%        | 62%        | 11%    |
|                      | Waterfall | 7%         | 68%        | 25%    |
| Small Size Projects  | Agile     | 58%        | 38%        | 4%     |
|                      | Waterfall | 44%        | 45%        | 11%    |

**Task complexity and durations (Rosato, 2018).**

PMI Survey results in the annual “Pulse of the Profession” 2018 report shows the top six reasons why project fail as deduced from the response of 3,000 different individuals in various industries.

**Figure 8: Reprinted from PMI's *Pulse of the Profession 9th Global Project Management Survey*, by Project Management Institute, 2017.**



## Conclusion and Recommendations

This paper points out that risks are inevitable in all software development projects hence risk management and decision making is vital. The review of existing literature establishes that risk assessment has been done qualitatively hence it is subjective. The paper points out that a more objective, repeatable, and observable risk assessment and decision-making support for risk management is necessary. Quantitative risk assessment analyzes the effect of those risk events by assigning a numerical rating to the risks. Decisions are based on established rules and utility function. Bayesian belief network (BBN) determines the causal factor that directly affect project outcomes by giving the correlations between risk factors and project outcomes. Normative Decision Theory model describes how individuals should make decisions. Expected utility theory optimizes decisions making under uncertainty. The rough set theory provides

efficient algorithms for finding hidden patterns in data and generate sets of decision rules from data. Decision rules derived from a decision table links conditions to particular actions, preferences or decisions. Decision networks which is an extension of a Bayesian networks represents a decision problem. Evaluation of the decision networks based on an algorithm returns the action with the highest utility.

## References

Chaos Manifesto (2013). Chaos Manifesto, Act Small. *Chaos Manifesto*, 2, 52. Retrieved From <Http://Www.Versionone.Com/Assets/Img/Files/Chaosmanifesto2013.Pdf>

Cunha, J. A. O. G., Da Silva, F. Q. B., De Moura, H. P., & Vasconcellos, F. J. S. (2016). Decision-Making In Software Project Management. In *Proceedings Of The 9th International Workshop On Cooperative And Human Aspects Of Software Engineering - CHASE '16* (Pp. 26–32). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2897586.2897598>

Hussain, A., Mkpojiogu, E. O. C., & Kamal, F. M. (2016). International Review of Management and Marketing the Role of Requirements in the Success or Failure of Software Projects. *International Review of Management and Marketing*, 6(S7), 11–13. Retrieved From <Http:Www.Econjournals.Com>

Kaur, R., & Sengupta, J. Ork (2011). Software Process Models and Analysis on Failure of Software Development Projects. *International Journal of Scientific & Engineering Research*, 2(2), 1–4. Retrieved From <Http://Arxiv.Org/Abs/1306.1068>

Kothari, C.R. (2004) Research Methodology: Methods and Techniques. 2nd Edition, *New Age International Publishers, New Delhi*.

Kumar, C., & Yadav, D. K. (2015). A Probabilistic Software Risk Assessment and Estimation Model for Software Projects. *Procedia Computer Science*, 54, 353–361.

<https://doi.org/10.1016/j.procs.2015.06.041>

PMI (2016). *The High Cost of Low Performance. Pulse of the Profession* (Pp. 1–20).

Project Smart (2014) . Exploring trends and developments in project management today

<https://www.projectsmart.co.uk/why-software-projects-fail.php>

Project Success Survey (2018) Driving project success in Belgium, 2018

[www.ambyssoft.com/surveys/success2018.html](http://www.ambyssoft.com/surveys/success2018.html)

Rekha, J. H., & Parvathi, R. (2015). Survey on Software Project Risks and Big Data Analytics. *Procedia Computer Science*, 50, 295–300.

<https://doi.org/10.1016/j.procs.2015.04.045>

Rosato, M. (2018). *Go Small for Project Success 1*, VII(V), 1–10.

Wanderley, M., Menezes, J., Gusmão, C., & Lima, F. (2015). Proposal of Risk Management Metrics for Multiple Project Software Development. *Procedia Computer Science*, 64, 1001–1009. <https://doi.org/10.1016/j.procs.2015.08.619>.