International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1509-1517 www.ijaem.net ISSN: 2395-5252

# Visualising Risk Levels, their Impacts and their Probability of Occurrence in the Requirement Elicitation Phase of Software Projects

Beatrice O. Akumba[1], Barnabas T. Akumba[2] Samera U. Otor[1], and Selumun Agber[1]

[1]*Department of Mathematics/Computer Science, Benue State University Makurdi, Nigeria*
[2] *Health Management Information Systems Programme (HISP) Abuja, Nigeria*
*Corresponding Author: Beatrice O. Akumba*

Submitted: 25-06-2021 Revised: 06-07-2021 Accepted: 09-07-2021

**ABSTRACT**
The requirement elicitation phase of software development is an important process in the software development life cycle (SDLC) as it entails an extensive and a thorough finding of information process gathered from all team members associated with the built of software. This phase of the software development process is associated with uncertainties and risks such that if the risks are visualised early enough, managing them will improve software quality and enhance timely deliverables of deadlines. Data visualisation is concerned with the task of putting data into a chart, graph or other visual formats that help inform analysis and interpretation. The widely used data visual formats are; frequency tables, cross-tabulation tables, bar charts, line graphs, pie charts, Heat maps, scatter graphs and histograms. The histogram data visual format has been considered and used in this paper to interpret the risk levels, their impacts and its probability across software projects for informed decision making by project managers and developers. The dataset for software risk prediction was adopted to visualise the risk levels and their impacts as well as their probability of occurrence in the software projects. This was necessitated to provide a pictorial view for risk decision support system and to educate the software developers on what to look out for as to foster better planning and timely project deliveries. A machine learning technique was applied to predict the software risk levels and their levels of impact which was represented using histograms in R. The Architecture of the Naïve Bayes Risk prediction system was also developed and an analysis of the performance metrics of the risk attributes with respect to the risk levels across the projects was also done using R. The Naïve Bayes Risk Prediction and Analysis Model (NBRPAM) had a plot of the performance matrix of the variables which showed that probability and priority are the most significant variables in predicting Risk levels in software projects.

**KEYWORDS:** Requirement Elicitation, Data Visualisation, Risk Levels, Software Development Life Cycle (SDLC), Histogram**,** Naïve Bayes, R programming

## I. INTRODUCTION

The requirement elicitation phase of software development is an important process in the software development life cycle. It entails an extensive and a thorough finding information process gathered from all team members associated with the built of software. [1]. This phase of the software development process is associated with uncertainties and risks and if the software risks are identified early enough, managing them will go a long way to improve software quality and enhance timely deliverables of deadlines. Risk is common and unavoidable in all the development stages of the software development lifecycle. Some measures and efforts have been used to mitigate risk in software projects but sometimes the risks are still persistent as no method has satisfactorily identified and managed the risk properly. This can be liken to the user requirement changes, technological changes, software evolvement and its complexities.

Data visualisation is concerned with the task of putting data into a chart, graph or other visual formats that help inform analysis and interpretation, [2]. The risk levels across software projects with respect to its impacts and probability have been equally interpreted using data visualisation. The widely used data visual formats are; frequency tables, cross-tabulation tables, bar charts, line graphs, pie charts, Heat maps, scatter graphs and histograms. The histogram data visual format has been considered and used in this paper

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1509-1517 www.ijaem.net ISSN: 2395-5252

to interpret the risk levels, their impacts and its probability across software projects for informed decision making by project managers and developers.

Impact can be thought of as the consequence or effect of a risk event on the software projects. The impacts are seen in their qualitative and quantitative manners [3]. A five point scale was used to measure the impact of risk on software projects using insignificant, low, moderate, high and catastrophic numerical scales across the projects.

Risk probability or likelihood is the probability of a risk event occurring. The probability assessment involves estimating the effects of a risk event on a set of software projects. Once risks are identified, its impact and probabilities are rated and the level assessed [4].

In this paper, the dataset for software risk prediction as developed by [5]. was adopted to visualise the risk levels and their impacts as well as their probability of occurrence in the software projects during the requirement gathering phase of the SDLC. This was necessitated to provide a pictorial view for risk decision support system and to educate the software developers on what to look out for as to foster better planning and timely project deliveries. Naïve Bayes classification algorithm was applied to predict the software risk levels and their levels of impact which was represented using histograms in R. The Architecture of the Naïve Bayes Risk prediction system was also developed and an analysis of the performance metrics of the risk attributes with respect to the risk levels across the projects was also done using R.

The rest of the paper is organized as follows: I Introduction, II Analysis of Risks in Software Process Models III Method, IV Results and Discussions and V Conclusion.

## II. ANALYSIS OF RISKS IN SOFTWARE PROCESS MODELS

[6]. reported that every phase of software development have certain uncertainties and assumptions in the software project. These uncertainties if not cleared and managed on time could give rise to software risks. They concluded that there exists severity of risks in all the stages of software development projects irrespective of the software model that was used for the development process. It was inferred that the severity of software project risks were high when the uncertainties were noticed in the earlier phases than at the later phases of software development.

**Table 1: Risk Severity at Various Phases (Bhujang and Suma, 2017)**

| Phases of Software Development | Severity of Risk |
|---|---|
| Analysis | 100% |
| Requirement Elicitation | 90% |
| Design/Prototyping | 70% |
| Coding/Building | 60% |
| Testing | 50% |
| Support/Maintenance | 30% |

[7] Reported that software project development process is associated with some levels of risks and are accompanied by high failure rates. In order to mitigate the risks, they proposed an intelligent model that was able to predict and manage the risks inherent in all software projects.

[8] Also emphasised that the growth in the complexity of software systems has made software performance predictions a difficult task. They addressed the problem using a model-based approach for resource utilization and software performance risk prediction. They employed machine learning techniques to predict the performance risk and resource utilisation.

[9]. reported the use of a machine learning based software risk assessment model using Naïve Bayes algorithm that risk management consisted of identifying, analyzing, planning, and controlling events that affect project environment. The model comprises the input selection process, data processing, classification of attributes, Implementation process using Naïve Bayes classification Algorithm and prediction of the risk process.

[5] developed a risk dataset that contained software requirements and risk attributes and also the relationship between requirement and risks, which is very cognizant in the prediction of risk in most software projects. It concluded that a dataset was necessary for research purpose to predict software risks at the requirement gathering stage. It is to this end that this study has adapted the dataset for visualizing the risk predicted using Naïve Bayes algorithm implemented in R as a means of getting first-hand information on attributes that pose risk in some software projects.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1509-1517 www.ijaem.net ISSN: 2395-5252

### II.I Importance Of Data Visualisation

Data visualisation is a graphical depiction of data which is usually represented using dashboards or reports. It illustrates the views of data that answer "what," such as, "What are the factors or attributes that pose risk in software projects? They are good for answering a finite set of questions, and can be static or provide some level of interactivity for investigating those questions. The important of visual analysis are: [10]

(i) A very good visual analysis platform enables you to easily create impactful visualisations and dashboards, and encourages exploration to identify new trends in the data of study.
(i) Supports forage freely in the data of interest by identifying outliers and reaching meaningful insights much faster.
(ii) Visual analytics supports self-directed, open-ended data exploration that allows thoughts to be followed visually down different paths
(iii) Data and visualisation should work in tandem. The steps of querying, exploring and visualising data should come together in a single process. Good visual analytics allows for fast exploration, iteration, prototyping and sketching with data to support the way the data is manipulated
(iv) It is very useful in predicting risks attributes in the future, for instance, the system should allow for predictive forecasting; when you want to understand a trend of occurrence across software projects, drag and drop trend lines should be available.

### II.II USE OF VISUALS IN ANALYSIS

Visual analytics applies the power of visual perception in order to allow humans participate in solving complex problems with data. The human involvement could be as a result of humans being better than machines in solving some tasks and because human understanding and interpretation is desired. Developing perceptually effective visual representations can enable fast, accurate, and trustworthy interpretation of machine learning models. There are two generic modalities that can be used for model interpretation with visual analytics and they are; Visualizing Model Structure otherwise called White-Box. Some examples are for the transparent models include decision trees, Random forest, Naive Bayes etc., where one option is to use visualization to represent the structure built through the training method. Several examples exist in this area, especially for decision trees and rules. The other one is Visualizing Model Behavior also known as Black-Box and does apply visualization as a way to look at the behavior of the model by understanding the relationship between input and output. [11]. This paper used the Visualizing Model Structure otherwise called White-Box structure to build a model in R by splitting the datasets into the training and testing instance using naive bayes algorithm.

### II.III USE OF R TO GENERATE THE VISUALS

Visualisation is a great example of how data visualisation can help decision makers in all organisations in which the software engineering community is not left out. It is usually implemented in R and other programming languages. R Programming offers a satisfactory set of inbuilt function and libraries (such as ggplot2, leaflet, lattice) to build visualisations and present data. The basic visualisation tools in R are; Histogram, Bar / Line Chart, Box plot and Scatter plots. The world today is filled with data and it becomes imperative that it is analysed properly to gain meaningful insights. Data Visualisation is a vital tool that can unearth possible crucial insights from data [12]. If the results of an analysis are not visualised properly, it will not be communicated effectively to the desired audience hence the study to visualise the risk levels and their impacts across the software projects data gathered.

## SECTION III: METHODS

In order to visualise the risk levels and their impacts across the software projects data gathered by [5] of the requirement gathering phase, Machine learning techniques have been used for the prediction and visualisation. Machine Learning techniques are powerful tools for software risk predictions. The supervised machine learning technique has been employed in this paper using the Naïve Bayes classifiers machine learning algorithm. Supervised learning is a learning model that is developed to make predictions based on an unforeseen input instance. A supervised learning algorithm accepts a known set of input dataset which are known responses to the data (output) to learn the classification/regression model. It is a learning algorithm that trains a model to produce a prediction for the response to new data or the test dataset. Supervised learning uses classification algorithms and regression techniques to develop predictive models. Classification task predicts discrete responses and it is recommended and applied for only data that can be categorized, tagged, or separated into specific groups or classes.

R programming language was used for the implementation and testing of the risk outcome prediction model. R programming is one of the

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1509-1517 www.ijaem.net ISSN: 2395-5252

promising languages for machine learning and data science as it provides excellent visualization features which are essential to explore the data before pushing it to any automated learning and assessing the results of the learning algorithm.

This paper has used Naïve Bayes to model the dataset of [5] in R to visualise the extent of risk and their corresponding impacts on the some selected software projects during the phase of requirement gathering of the SDLC. The dataset used the architecture of the system and the graphs are presented in the subsequent sub-sections as follows:

**a. The Architecture of the System**

The system architecture is shown in Figure 1 and it explains the working of the system. Risk management consists of identifying, analyzing, planning and controlling events that could cause problems in software projects environment. The system architecture is designed to identify and analyze the attributes that contribute to the risks and also, predict the risk levels which are grouped into five categories on the basis of priority. the Input selection is the first process which is the datasets collected on five different projects during the requirement gathering stage of the software projects. It is followed by the Data pre-processing method which cleans the datasets of errors and discrepancies and eliminating unwanted value or symbols or characters in the dataset. The Attribute selection is done to select the attributes in the dataset to be fed into the Naïve Bayes model for training. This is followed by the model testing and the risk outcome predictions.

**b. The Dataset for the Model**

The dataset used for the prediction study was adapted from the Dataset Software Requirement Risk Prediction from https://doi.org/10.5281/zenodo.1209601 [5]. The dataset has 299 data instances for risk prediction at the initial phase of the software development lifecycle (requirement gathering phase). The dataset consist of thirteen (12) variables attributes and one (1) target variable (outcome risk).

**c. The Algorithm of the Risk Prediction and Visualization System**

The algorithm of the system is described using a Pseudo code as shown below:

**Begin**
Load R Packages (tidyverse, ggplot2, caret, caretEnsemble, psych, Amelia, mice, GGally, rpart, randomForest) needed for the prediction
Read the Dataset file into R

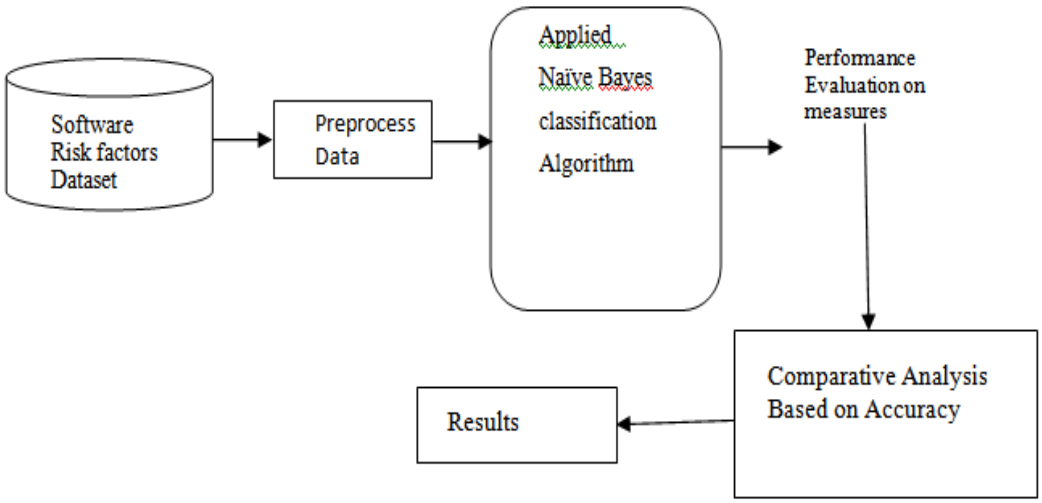


**Figure 1: The Architecture of the System**

**Set outcome according to the risk levels**
**Split the dataset into two partitions of 75% training and 25% testing instances**
**Let objects x hold predicted variables and object y hold response variables**
**Load the Naïve Bayes Function Library (e1071)**
**Create Naïve Bayes Model using the training dataset**
**Perform prediction using the test instances**
**Print output of the predictions using ggplots**
**End**

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1509-1517 www.ijaem.net ISSN: 2395-5252

**d. Implementation**

In this paper, R programming language (RGui - 64 bit) was used for the implementation and testing of the Naïve Bayes Model. R programming language is one of the major languages for machine learning and data science. It provides excellent visualisation features, which is essential to explore the data before submitting it to any automated learning, as well as assessing the results of the learning algorithm.

## IV. RESULTS AND DISCUSSIONS

The results of the risk prediction model were analysed and visualised in terms of the impact, magnitude, priority (catastrophic, high and low), and the probability of the occurrence of the risk. The results of the analysis of each of the parameters are discussed in detail with their respective plots showing the behavior of each outcome.

**(i) The Dataset Partition (Training and Testing)**

The dataset for the model was split into two phases of train and test instances. 75% of the dataset was used for training the Naïve Bayes model while 25% of the dataset was used for the testing and validation of the model.

**(ii) The Naïve Bayes Risk Prediction and Analysis Model (NBRPAM)**

A Naïve Bayes Model for the prediction and analysis of the risk in the software projects was created. The model has 226 dataset samples corresponding to the 75% of the dataset that was partitioned for the model training. The Model has 12 predictors corresponding to the attributes. Ten-fold (10-fold) Cross-Validation was applied and the false and true accuracies stood at 98% and Kappa was 97%.

**(iii) Visualising the Risk levels across the Projects as generated using R**

In order to understand the risk levels across the software projects, ggplots histograms were used to visualise the risk levels in terms of their impact, magnitude, priority (catastrophic, high and low), and the probability of the occurrence of the risk across the software projects for easy visualisation and interpretation.

**(a) Visualising the Risk levels according to its Impact:** Impact can be thought of as the consequence or effect of a risk event on the software projects. The impacts are seen in their qualitative and quantitative manners. A five point scale is used to measure the impact of risk on software projects. They are; insignificant, low, moderate, high and catastrophic but often defined using numerical scales. Figures 2 and 3 are results of ggplots carried out to visualise the relationship between impact and the levels of risk (Catastrophic, High, Moderate, Low and insignificant) across the projects. The ggplot in Figure 3 shows that risks categorized as moderate and low had more impact on the project. For all the levels of risk, the impact increased with the counts of the risk levels. This shows that the number of risks (irrespective of the level of categorisation) has serious impact on the project.

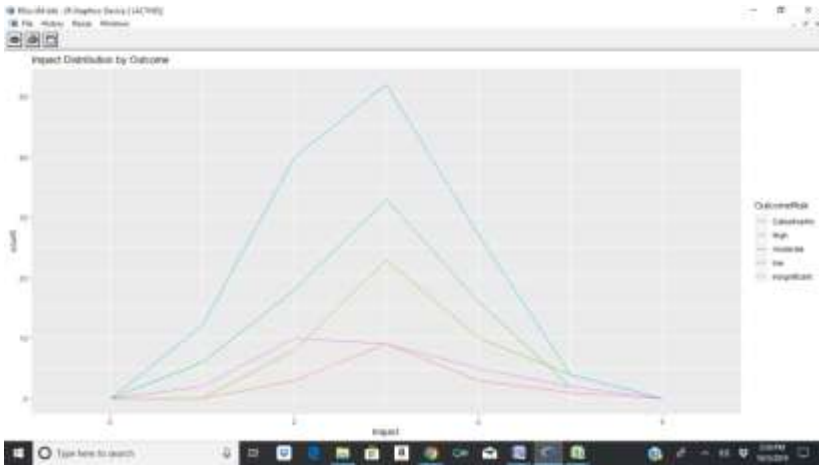


**Figure 2: The Impact Distribution by Outcome (Risk Levels)**

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1509-1517 www.ijaem.net ISSN: 2395-5252

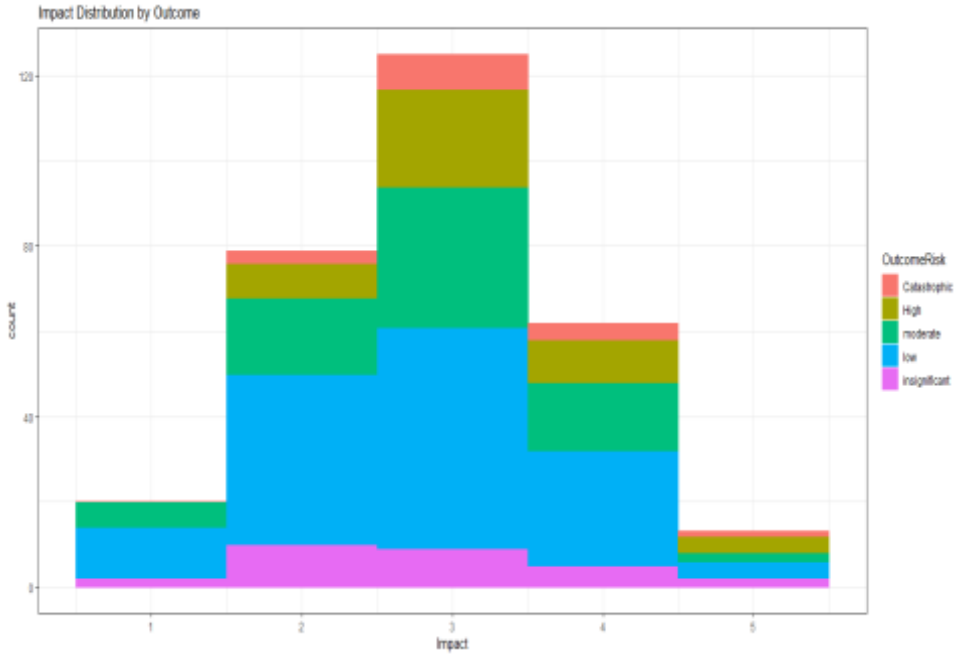


**Figure 3: Impact Distribution by Outcome (Risk Levels)**

**(b) Visualising the Risk levels according to the Magnitude Distribution by Outcome**

Magnitude is a measure of the probability of occurrence of a risk and the severity of its consequences. The probability score is a measure of the likelihood of occurrence of the risk scenario, and the severity score is a measure of the amount of damage or penalty to be expected. Figure 4 shows the result of a ggplot of magnitude against the risk levels of the project. It can be seen that for all the levels of risk (outcomes) the numbers increased with an increase in the magnitude. It can be seen that an increase in the number (counts) of risks categorized as low increased the magnitude. The maximum number of low level risks was recorded at a magnitude of 2

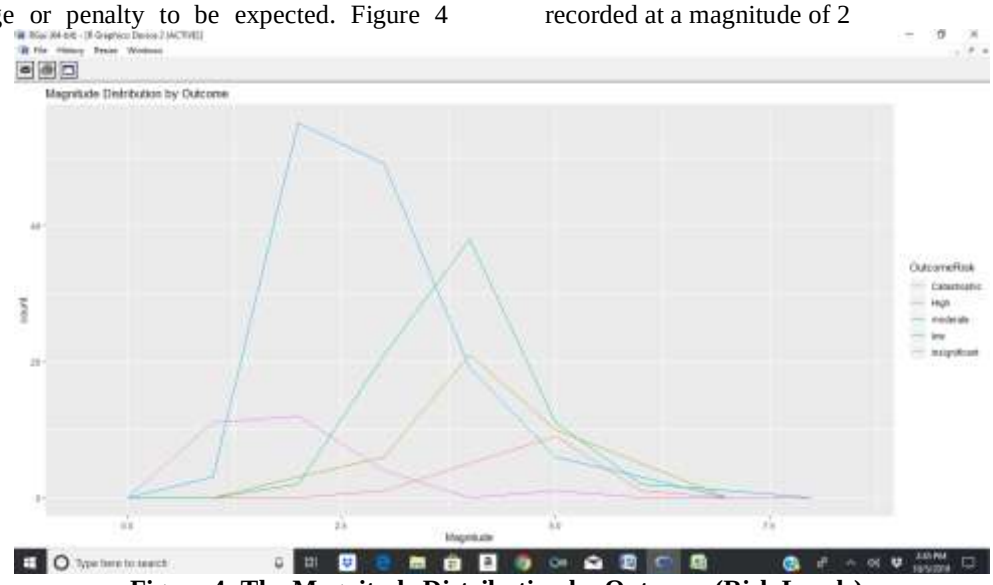


**Figure 4: The Magnitude Distribution by Outcome (Risk Levels)**

**(c) Visualising the Risk levels according to the Priority Distribution by Outcome (Risk Levels)**

Priority is the order in which the risk levels are to be resolved. It can be marked as low, medium, high, and urgent. Figure 5 is a ggplot histogram of priority against the risk levels. The ggplot shows that higher level risks have more priority and as such should be resolved as early as possible to avoid the consequences of the risks on the projects

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1509-1517 www.ijaem.net ISSN: 2395-5252

life cycle. The order of priority as shown in plot is; Catastrophic, High, Moderate, Low and Insignificant.

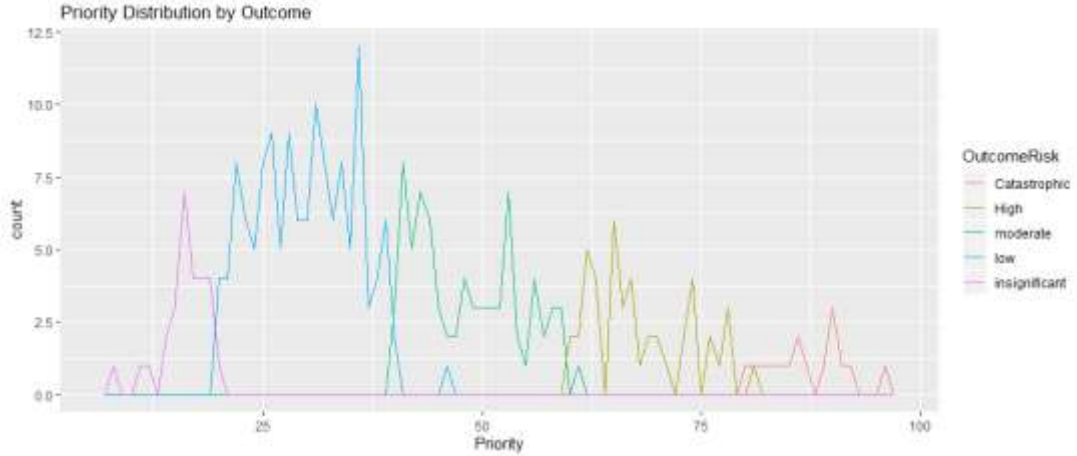


**Figure 5: Risk Priority Distribution by Outcome (Risk Levels)**

**(d) Visualising the Risk levels according to the probability Distribution by Outcome**

Figure 6 shows the result of the probability distribution by risk levels. The risk levels increased as probability increased. The numbers (counts) of risks within the low risk category were higher at the lower probability levels. There was a considerable increase in the number (counts) of risks under the catastrophic level at higher priorities and lower increase levels at lower probability**.**

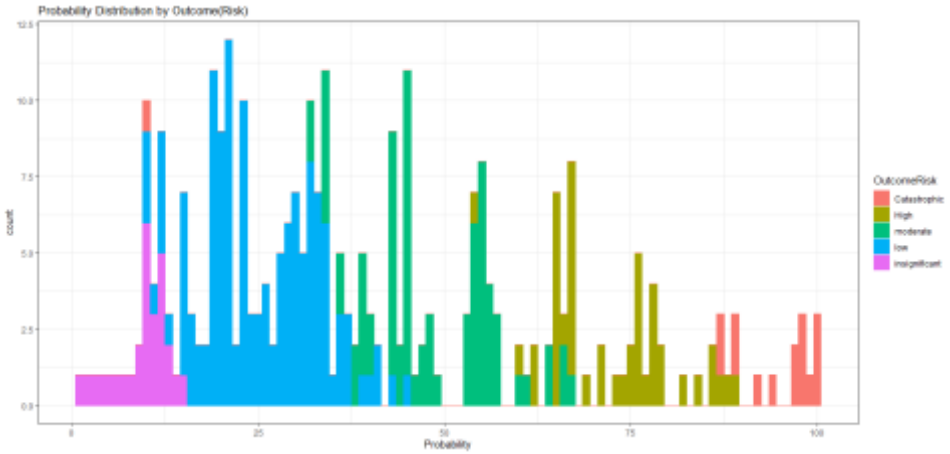


**Figure 6: Result of the Probability Distribution by Outcome (Risk Levels)**

**(e) Visualising the Variable Performance across the Projects**

Figure 7 shows the plot of the performance matrix of the variables. The plot shows that priority should be given to all the levels of risk as they have the ability to affect the outcome of the project. The plot also revealed that there is high probability of all levels of risk to occur during the project life cycle. Software project managers should put in place measures to reduce the probability of the occurrence of risks and prompt measures to eliminate the various risks when they occur

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 1509-1517 www.ijaem.net ISSN: 2395-5252

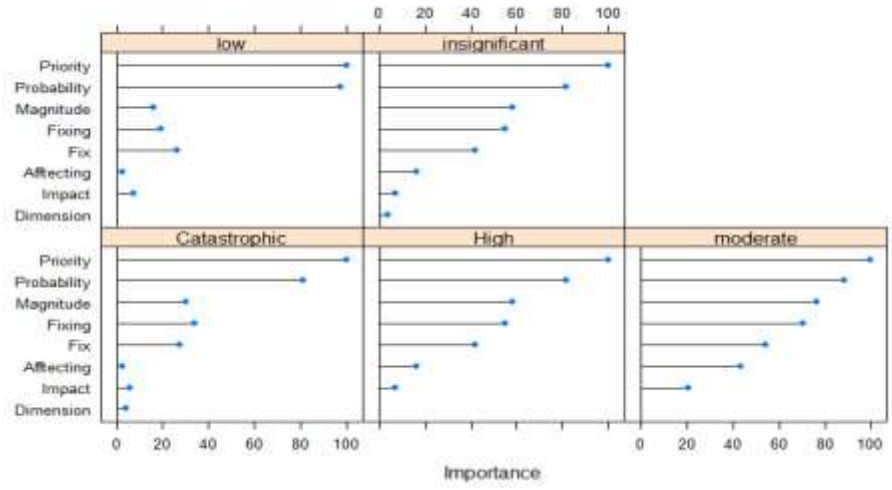


**Figure 7: Plot of the performance matrix of the variables across the Project**

## V. CONCLUSION

The requirement elicitation phase of software development is an important process in the software development life cycle. It entails an extensive and a thorough finding information process gathered from all team members associated with the built of software. This phase of the software development process is associated with uncertainties and risks and if the software risks are identified early enough, managing them will go a long way to improve software quality and enhance timely deliverables of deadlines. Risk is common and unavoidable in all the development stages of the software development lifecycle. Data visualisation is concerned with the task of putting data into a chart, graph or other visual formats that help inform analysis and interpretation. The risk levels across software projects with respect to its impacts and probability have been equally interpreted using data visualisation. The widely used data visual formats are; frequency tables, cross-tabulation tables, bar charts, line graphs, pie charts, Heat maps, scatter graphs and histograms. The histogram data visual format has been considered and used in this paper to interpret the risk levels, their impacts and its probability across software projects for informed decision making by project managers and developers. Naïve Bayes classification algorithm was applied to predict the software risk levels and their levels of impact which was represented using histograms ggplots in R. The Naïve Bayes Risk Prediction and Analysis Model (NBRPAM) had a plot of the performance matrix of the variables which showed that probability and priority are the most significant variables in predicting Risk levels in software projects. Therefore, Software Project Managers should put in place measures to reduce the probability of the identified risks.

## REFERENCES

[1]. Ramdhani, A., Sa'adillah M., Maylawati, D. Amin, A., and Aulawi, H. (2018). Requirements Elicitation in Software Engineering. International Journal of Engineering & Technology, 7(2.29), 772. doi:10.14419/ijet.v7i2.29.14254

[2]. Mealdprostarter (2019). [online] https://mealdprostarter.org/n-data-analysis-visualization-and-interpretation/

[3]. Karlotta, T. (2018). Impact and Probability in Risk Assessment [online] http://apppm.man.dtu.dk/index.php/Impact_and_Probability_in_Risk_Assessment#cite_note-Curtis-2

[4]. Curtis, P. and Carey, M. (2012) Risk Assessment in Practice. Deloitte & Touche LLP.

[5]. Shaukat, Z., Naseem, R. and Zubar, M. (2018). A Dataset for Software Requirement Risk Prediction. IEEE International Conference on Computational Science and Engineering. Pp. 112-118

[6]. Bhujang, R.K. and Suma, V. (2017). Recent Trends of Risk Management in Software Development: An Analysis. International Scientific Journal of Contemporary Research in Engineering, Science and Management (ISJCRESM), Vol. 1, Issue 4, pp 33-43.

[7]. Hu, Y., Feng, B., Mob, X., Zhang, X. Z., Ngai, E. W. T., Fan, M. and Liu, M. (2015). Elsevier Journals on Decision Support Systems. 72, 11-23 [online]. Retrieved on 9th August, 2019 from

https://www.sciencedirect.com/science/article/pii/S0167923615000238

[8]. Salih, H. A. M. and Amar, H.H. (2017). Model-Based Resource Utilisation and Performance Risk Prediction Using Machine Learning Techniques. International Journal on Informatics Visualisation (JOIV). 1(3), p 101-109.

[9]. Suresh, K. and Dillibabu, R. (2018). Designing a Machine Learning Based Software Risk Assessment Model using Naïve Bayes Algorithm. TAGA Journal, Vol.14

[10]. Mathew, N. (2017). Tableau White Paper. Why Visual Analytics? Accessed September 2020 online from https://cdn2.hubspot.net/hubfs/2383378/Tableau%20Whitepaper%20-%20Why%20Visual%20Analytics.pdf?t=1520904633993

[11]. ] Krause, J., Perer, A. and Bertini, E. (2016). Using Visual Analytics to Interpret Predictive Machine Learning Models. ICML Workshop on Human Interpretability in Machine Learning (WHI 2016), New York, NY, USA.

[12]. Pandey, P. (2019). An overview of the R visualisation capabilities. A Comprehensive Guide to Data Visualisation in R for Beginners. [online] https://towardsdatascience.com/a-guide-to-data-visualisation-in-r-for-beginners-ef6d41a34174#c517