A Dataset for Software Requirements Risk Prediction

Zain Shaukat
Department of Computer Science
City University of Sciences and Information Technology
Peshawar, Pakistan
zsadozai22@gmail.com

Rashid Naseem
Department of Computer Science
City University of Sciences and Information Technology
Peshawar, Pakistan
rashid@cusit.edu.pk

Muhammad Zubair
College of Electrical and Mechanical Engineering
National University of Sciences & Technology
Islamabad, Pakistan
zubair.ceme@gmail.com

***Abstract*— The risk prediction in the software development is mandatory for it to be recognized, categorized and prioritized earlier for the success of the project. The requirement gathering stage is the most important and challenging stage of the Software Development Life Cycle (SDLC). The risks should be tackled at this stage and saved it to be used in future projects. The software requirement risks can be predicted using classification techniques of data-mining at requirement gathering stage. A dataset is required containing the attributes of software requirements and risks for the prediction of risks in the new software requirements. In this paper, a risk dataset is proposed which contains requirements from the Software Requirement Specification (SRS) of different open source projects and the risk attributes from literature and IT experts. The research comprised of three main phases that include risk-oriented data collection, dataset validation by IT experts, and dataset validation and filtration.**

Keywords*— *Software Risk, Software Development Life Cycle (SDLC), Data-mining, Software Requirement Specification (SRS), Dataset, Dataset Preprocessing.

I. INTRODUCTION

There is always a chance of uncertain events in the process of SDLC which may lead to potential loss of software development or software organization. These uncertain events are called software risk. The risks hail from different risk factors which are rooted in a variety of activities in the project development life cycle. If these risks are not identified properly they may become responsible for the failure of the project [1, 2, 3, 4]. These factors need to be identified and mitigate to minimize the software cost and schedule by the prediction of risks in the initial stages of the software development lifecycle. Requirement gathering is the initial step of SDLC, therefore prediction of risks at this stage can improve the quality and the efficiency of a software and will reduce the chances of failure of the project. Numerous methods for software risk prediction at several stages in SDLC are available so far. To the best of our knowledge, rare techniques exist to predict risks at the requirement gathering phase [2, 3].

Previously in literature, Salih and Ammar [3] used machine learning techniques for the software performance risk prediction. Moreover, Purandare [5] proposed an entropy-based approach for the analysis of risk factors of the software projects using Cocomo-sdr dataset. Christiansen et al [6] have performed Logistic Regression on software development projects to predict risks on the basis of the questionnaire from experts. Analytical Hierarchal Processing (AHP) has been used by Fang and Marle [7] to identify risks and risk interactions. Nassif et al had analyzed nonfunctional requirements using Multiple Linear Regression model and Artificial Neural Network (ANN) in the Desharnais dataset for Software Effort Estimation [13].

In the mentioned literature the datasets used by Pradnya Purandare and Nassif et al are related to effort estimation. However, these datasets contain attributes for effort estimation while they do not have the risk attributes like risk category, magnitude of risk, impact, dimension of risk, probability and priority as described by Laurie Williams [4] and Boehm [8]. Since none of them has presented a requirement risk dataset which contains risk-oriented attributes for the prediction of risk in software requirement gathering phase. The details of the techniques and datasets used by authors are shown in Table I.

TABLE I. TECHNIQUES AND DATASETS USED BY AUTHORS

Authors	**Data source**	**Technique**	**Domain**
Pradnya Purandare	Cocomo sdr [19]	entropy-based approach	Risk Factor Analysis
Christiansen et al	Experts Questionnaire	Multiple Logisitic Regression	Risk Factors Prediction
Fang and Marle	Interview	AHP	Network-Based Risk Identification
Salih and Amar	Hospital Management system	Naïve Bayes, SVM,	Performance Risk Prediction
Nassif et al	Desharnais Dataset [20]	ANN	Non-Functional Requirements

As far, no dataset containing risk attributes for software requirements currently exists. Therefore, in this paper risk dataset is proposed that contains software requirement and risk attributes and the relation between requirements and risks, which are necessary for the prediction of risks in the upcoming software projects. The dataset makes it able to predict risks over

978-1-5386-7649-3/18/$31.00 ©2018 IEEE
DOI 10.1109/CSE.2018.00022

the SRS of a new project using classification techniques of data mining. These techniques can be simulated using data mining tool like Waikato Environment for Knowledge Analysis (WEKA) [10]. Weka is a free software having a collection of machine learning algorithms for data mining tasks. The algorithms can be applied directly to the dataset [3, 10].

The rest of this paper is organized as follows:- Section II presents the proposed methodology of the paper. Section III consists of experimental results. In Section IV the conclusion of this work is discussed.

II. Proposed Methodology

This section is divided into three main phases that are explained in detail.

A. Risk Oriented Data Collection

Data is collected through different SRS of different open source projects that include,

- Transaction Processing System [15], It contains the software requirements specification related to a web-based student information management system. This SRS has 118 requirements collectively.
- Management Information System [16], That contains the E-Store product features. It focuses on the company, the stakeholders, and applications, which allow for online sales, distribution, and marketing of electronics. This SRS collectively contains 87 Requirements.
- Enterprise System [17], This SRS identifies requirements focused on medical records and the associated diagnostics related to web-based Patient, Physician, and Ambulatory Input/Outputs, and sensor driven inputs for real-time patient monitoring. This SRS has 59 requirements.
- Safety Critical System [18], The intelligent Traffic Expert Solution for road traffic control System offers the ability to acquire real-time traffic information. Traffic Expert enables operators to perform real-time data analysis on the information gathered. It has 35 requirements.

The dataset contains 299 instances (Requirements) collectively as mentioned in Table II.

TABLE II. Projects And Number Of Instances

Project	**Instances**
Transaction Processing System	118
Management Information System	87
Enterprise System	59
Safety Critical System	35
Total	**299**

Requirement risk dataset can be downloaded from [14]. The dataset is formed by adding risk attributes to the SRS data. The risk attributes project category, requirement category, risk target category, probability, impact, dimension of risk, and priority of risk were collected from [1, 4, 8]. Moreover, other attributes included by the IT experts form CMMI Level 2 and CMMI Level 3 of the IT Industry. Those IT experts have more than five years of experience in the field. The attributes are affecting number of modules, cost of risk and fixing duration which is commonly used in the industry for risk assessment as shown in Table III.

TABLE III. Requirement Risk Attributes Sources

Attributes	**Source**
Project Category	SRS [15, 16, 17, 18]
Requirement Category	SRS [15, 16, 17, 18]
Risk Category	[1, 4, 8, 22]
Magnitude of Risk	[1, 4, 8]
Impact	[1, 4, 8]
Dimension of Risk	[1, 4, 8]
Probability	[1, 4, 8]
Priority	[1, 4, 8]
Affecting No of Modules	IT Experts
Fixing Duration	IT Experts
Fix Cost	IT Experts
Risk Level	Result

The whole formation is setup into Excel datasheet format as shown in Fig 1.

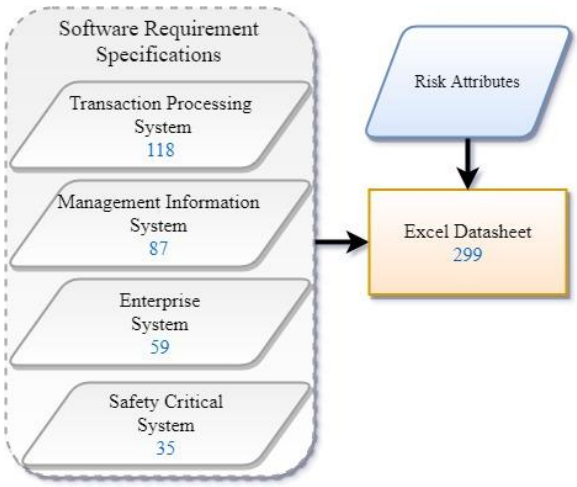

Fig 1. Risk data collection

B. Risk validation by IT experts

The excel datasheet containing the requirement risk attributes are filled by IT experts, on the basis of their experience in the IT industry as shown in Fig 2.

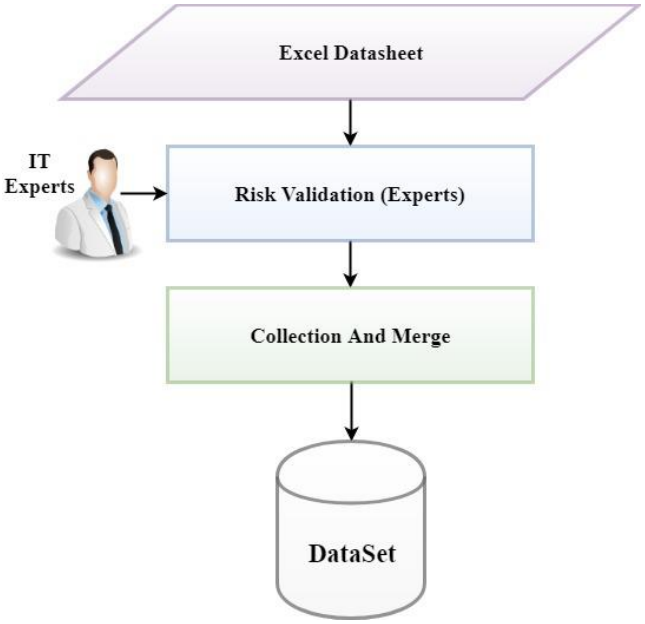

Fig 2. Risk dataset validation by IT Experts

After collection of risk measures from IT experts, the excel datasheet is grouped and molded in the form of Comma Separated Values (CSV). CSV is a readable format for WEKA.

The data types of the attributes were assigned according to nature and value set of the data which were achieved from Boehm's risks management [1, 4, 8] and IT Experts. The attribute Requirements is assigned to String data type. The other attributes i.e., project category, requirement category, risk category, magnitude of risk, impact, dimension of risk and risk level were assigned with a nominal data type because of categorical values [4]. The attributes probability, affecting number of modules, fixing duration, fix cost and priority are set to numeric data type. The attributes, their data types and their values are shown in Table IV.

TABLE IV. RISK DATASET TEMPLATE

Attributes	**Data types**	**Value Sets**
Requirements	String	Text
Project Category	Nominal	Transaction Processing System, Safety Critical System, Enterprise System, Management Information System,
Requirement Category	Nominal	Functional, Usability, Reliability & Availability, Performance, Security, Supportability, Constraints, Interfaces, Standards, Safety
Risk Target Category	Nominal	Budget, Quality, Schedule, Personal, Performance, Functional Validity, People, Project complexity, Planning & Control, Team, Resource availability, User, Requirement, Time Dimension, Organizational Environment, Cost, Design, Business, Unrealistic Requirements, Overdrawn Budget, Software, Process
Probability	Numeric	Percent 0-100
Magnitude of Risk	Nominal	Negligible, Very Low, Low, Medium, High, Very High, Extreme
Impact	Nominal	high, catastrophic, moderate, Low, insignificant
Dimension of Risk	Numeric	Requirements, User, Project complexity, planning and control, Team, Organizational Environment, Estimations, Software Requirement, Planning and Control, Schedule, Complexity, Project cost, Organizational Requirements
Affecting No Modules	Numeric	Numbers
Fixing Duration	Numeric	Days
Fix Cost	Numeric	Percentage of the total project cost
Priority	Numeric	Percent 0-100
Risk Level	Nominal	1, 2, 3, 4, 5

The Risk Level values were grouped in five ranks on the basis of priority of the risk. In the last phase, the dataset is converted and saved in Attribute-Relation File Format (ARFF) [14]. Some instances from the dataset are shown in Table V, that illustrates risks occurrence in the requirements.

TABLE V. REQUIREMENT RISK DATASET ARFF FORMAT

Requirements	project Category	Requirement Category	Risk Target Category	Probability	Magnitude of Risk	Impact	Dimension of Risk	Affecting No of Modules	Fixing Duration	Fix Cost	Priority	Risk Level
The system shall display all the products that can be configured	Transaction Processing System	Functional	Budget	10	Negligible	high	Requirements	9	1	10	95.7154	5
The system shall allow user to select the product to configure	Transaction Processing System	Functional	Quality	22	Very Low	catastrophic	Requirements	7	2	11	35.9	2
The system shall display all the available components of the product to configure	Transaction Processing System	Functional	Schedule	33	Low	high	User	5	1	3	35.6923	2
The system shall provide a uniform look and feel between all the web pages	Transaction Processing System	Usability	Quality	64	Medium	high	Planning and Control	2	3	1	57.35	3
The user-0.25 screen shall respond within 5 seconds	Management Information System	Performance	Project complexity	29	High	Insignificant	Organizational Requirements	8	4	1	36.16	2
Smart Traffic Management System can help by creating awareness	Safety Critical System	Functional	Requirement	16	Very Low	Moderate	Project Complexity	1	2	0	20.33	2
The system will provide employees with a login.	Enterprise System	Functional	Requirement	12	Very Low	High	Planning and control	2	2	0	17.3	1

C. Dataset Validation and Filtration

KNN classification technique is used for the proposed dataset because KNN is a "Lazy Learner" and performs instance-based learning, a well-tuned K can model complex decision spaces having arbitrarily complicated decision boundaries, which are not easily modeled by other "eager" learners like Decision Trees, Bayesian, and Support Vector Machine [21]. It is also better in terms of accuracy compared to Decision Tree and Naïve Bayes algorithms for a software risk related environment consisting of nominal and numeric data [3]. This classifier is selected on the bases of literature for its better accuracy in the environment of risk related data [3, 9, 21]. A distance of point X to point Y is calculated by equation (1),

$$\mathrm{dist}(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

The preprocessing filters are also applied to get better accuracy on the proposed dataset. Three techniques that are used for preprocessing are Normalize, Standardize and Discretize from unsupervised learning preprocessors of WEKA [10]. These filters are explained in detail.

- Normalize: Normalizes all numeric values in the dataset. The result values are by default in 0-1 for the data used to calculate the normalization intervals. But with the scale and translation parameters one can change that, e.g., with scale = 2.0 and translation = -1.0 you get values in the range (-1, +1) [11, 12]
- Standardize: Standardizes all numeric attributes in the given dataset to have zero mean and unit variance also known as z-score Normalization. Mathematically it is shown in equation (2),

$$\mathbf{Z} = (\chi - \mu) / \sigma \quad (2)$$

  where χ is value, μ is mean and σ is standard deviation [12].
- Discretize: An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes. Discretization is by simple binning [10].

The process of filtration, classification, and comparison of results accuracy among filters is shown in Fig 3.

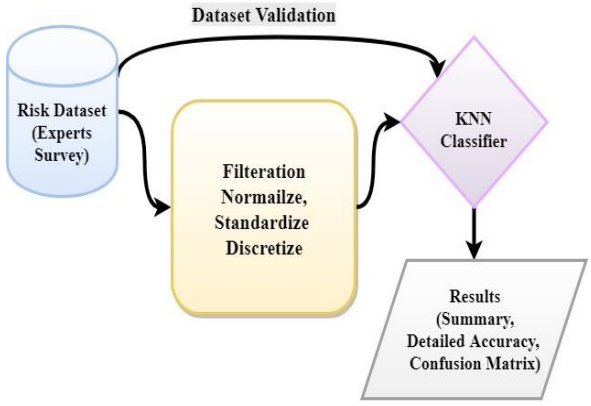

Fig 3. Model for validation and filtration of proposed dataset

III. Experimental Results

In this section, the dataset is validated using a KNN classifier. Similarly, three preprocessing filters are evaluated on the dataset, to achieve the best suitable preprocessing filter according to the nature of the proposed dataset. The validation results and the comparison of accuracy between preprocessing filters of the proposed dataset are mentioned in the subsequent section.

A. Dataset validation using KNN

In the first test, the proposed dataset is validated by 10 folds cross validation using the KNN classifier. The experimental results Show the correct class identification, incorrect class identification, Mean Absolute Error (MAE) shown in equation (3),

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \quad (3)$$

Root Mean Squared Error (RMSE) shown in equation (4),

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \quad (4)$$

Results are mentioned in Table VI, and the Confusion Matrix of the classification is shown in Table VII.

TABLE VI. Dataset Cross Validation Using Knn

KNN Classifier	Cross validation
Correctly Classified Instances	58.2%
Incorrectly Classified Instances	41.8%
MAE	0.17
RMSE	0.4052
Total Number of Instances	174/299

TABLE VII. Confusion Matrix Of Classification

Confusion Matrix					
A	**B**	**C**	**D**	**E**	**Classified As**
12	13	3	0	0	**A=1**
12	**96**	21	4	2	**B=2**
1	22	**41**	9	2	**C=3**
0	4	13	**22**	6	**D=4**
0	2	5	6	**3**	**E=5**

B. Using KNN with normalize filter

In the second test, we evaluated the dataset with Normalize filter and then cross validated using KNN 10 folds. Results of correct class identification, incorrect class identification, MAE, RMSE and the total no of instances are shown in Table VIII, and Confusion Matrix is shown in Table IX.

TABLE VIII. EXPERIMENTAL RESULTS USING KNN WITH NORMALIZE FILTER

KNN Classifier	Normalize
Correctly Classified Instances	58.2%
Incorrectly Classified Instances	41.8%
MAE	0.17
RMSE	0.4052
Total Number of Instances	174/299

TABLE IX. CONFUSION MATRIX OF CLASSIFICATION USING NORMALIZE DATASET

Confusion Matrix					
A	**B**	**C**	**D**	**E**	**Classified As**
12	13	3	0	0	**A=1**
12	**96**	21	4	2	**B=2**
1	22	**41**	9	2	**C=3**
0	4	13	**22**	6	**D=4**
0	2	5	6	**3**	**E=5**

C. Using KNN with standardize filter

In this test Standardize filtration for the proposed dataset, and then cross validated using KNN 10 folds. Results of correct class identification, incorrect class identification, MAE, RMSE and the total no of instances are shown in Table X and Confusion Matrix is shown in Table XI.

TABLE X. EXPERIMENTAL RESULTS USING KNN WITH STANDARDIZE FILTER

KNN Classifier	Standardize
Correctly Classified Instances	58.2%
Incorrectly Classified Instances	41.8%
Mean absolute error	0.17
Root mean squared error	0.4052
Total Number of Instances	174/299

TABLE XI. CONFUSION MATRIX OF CLASSIFICATION USING STANDARDIZE DATASET

Confusion Matrix					
A	**B**	**C**	**D**	**E**	**Classified As**
12	13	3	0	0	**A=1**
12	**96**	21	4	2	**B=2**
1	22	**41**	9	2	**C=3**
0	4	13	**22**	6	**D=4**
0	2	5	6	**3**	**E=5**

D. Using KNN with discretize filter

In the fourth test, discretize filter is applied to the proposed dataset then cross validating using KNN 10 folds. Results of correct class identification, incorrect class identification, MAE, RMSE and the total no of instances are shown in Table XII and Confusion Matrix is shown in Table XIII.

TABLE XII. EXPERIMENTAL RESULTS USING KNN WITH DISCRETIZE FILTER

KNN Classifier	Discretize
Correctly Classified Instances	76.25%
Incorrectly Classified Instances	23.74%
MAE	0.1056
RMSE	0.2798
Total Number of Instances	228/299

TABLE XIII. CONFUSION MATRIX OF CLASSIFICATION USING DISCRETIZE DATASET

Confusion Matrix					
A	**B**	**C**	**D**	**E**	**Classified As**
17	10	1	0	0	**A=1**
13	**114**	6	2	0	**B=2**
2	18	**51**	1	3	**C=3**
0	2	7	**36**	0	**D=4**
1	1	3	1	**10**	**E=5**

Experimental results reveal that Normalize and Standardize filters made no improvement in the accuracy results of KNN for the proposed dataset, while Discretize filter made 76.25% Correctly Classification of the instance which is 17% more than other filters. It also has a relatively low MAE that is 0.11 and RMSE of 0.280. The details comparison is presented in Table XIV.

TABLE XIV. COMPARISON OF PREPROCESSING FILTERS FOR THE PROPOSED DATASET

KNN Classifier	Without Filter	Normalize	Standardize	Discretize
Correctly Classified Instances	58.2%	58.2%	58.2%	76.25%
Incorrect Classified Instances	41.8%	41.8%	41.8%	23.74%
MAE	0.17	0.17	0.17	0.11
RMSE	0.405	0.405	0.405	0.280
Number of Correct instances / Total Instances	174/299	174/299	174/299	228/299

IV. Conclusion

A dataset is necessary mainly for the research purpose to predict software risks at the requirement gathering stage. For this purpose, a dataset is proposed that contains requirements data from four different SRS sources that are then validated by IT experts. The risks of the upcoming project can easily be predicted using different machine learning techniques for the proposed dataset. We used a KNN classifier for the validation of the dataset. Also, we have applied three preprocessing filters for the improvement of the accuracy of the proposed dataset.

The proposed dataset contains mostly categorical data because of which Normalization and Standardization resulted in the same accuracy. While Discretize converts the numeric attributes into ordinal range values, which results in classification more precise in terms of classification accuracy. According to results, we suggest Discretize filter for preprocessing because it has better performance for the proposed dataset.

The main aim for the proposed dataset is to provide a data source for risk decision support system for the improvement of risk prediction at the initial phase of the software development life cycle. The proposed dataset can also be used to support different areas of software project development like requirements and risks prioritization, software risk prediction, cost estimation, and effort estimation.

References

[1] T. T. Moores, R. Champion, and K. Tong, "A Methodology for Measuring the Risk Associated With a Software," Australas. J. Inf. Syst., vol. 4, no. 1, pp. 55–63, 1996.

[2] K. Appukkutty, H. H. Ammar, and K. G. Popstajanova, "Software requirement risk assessment using UML," 3rd ACS/IEEE Int. Conf. on computer Syst. Appl. 2005., pp. 1–4, 2005.

[3] A.. S. Haitham and H. H. Ammar, "Model-Based Resource Utilization and Performance Risk Prediction using Machine Learning Techniques," Int. J. Informatics Vis., vol. 1, no. 3, pp. 101–109, 2017.

[4] L. Williams, "Project Risks Product-Specific Risks," in Journal of security NCSU, vol. 1, no. 1, 2004, pp. 1–22.

[5] P. Purandare, "An Entropy Based Approach for Risk Factor Analysis in a Software Development Project," Int. J. Appl. Eng. Res., vol. 11, no. 4, pp. 2258–2262, 2016.

[6] T. Christiansen, P. Wuttidittachotti, S. Prakancharoen, and S. A. Vallipakorn, "PREDICTION OF RISK FACTORS OF SOFTWARE DEVELOPMENT," Asian Res. Publ. Netw., vol. 10, no. 3, pp. 1324–1331, 2015.

[7] C. Fang and F. Marle, "A Simulation-Based Risk Network Model for Decision Support in Project Risk Management," Elsevier Sci. Publ. B. V., vol. 52, no. 3, pp. 635–644, 2012.

[8] B. Boehm, "Software risk management: principles and practices," IEEE Softw., vol. 8, no. 1, pp. 32–41, 1991.Df

[9] V. Bijalwan, V. Kumar, P. Kumari, and J. Pascual, "KNN based Machine Learning Approach for Text and Document Mining," Int. J. Database Theory Appl., vol. 7, no. 1, pp. 61–70, 2014.

[10] T. C. S. and E. Frank, "Statistical Genomics: Methods and Protocols," in *smith16:_statis_genom*, Springer, 2016, pp. 353–378.

[11] J. Brownlee, "How to Normalize and Standardize Your Machine Learning Data in Weka - Machine Learning Mastery," *Machine Learning Mastery*, 2016. [Online].Available:https://machinelearningmastery.com/normalize-standardize-machine-learning-data-weka/. [Accessed: 17-Feb-2018].

[12] J. Brownlee, "How to Transform Your Machine Learning Data in Weka - Machine Learning Mastery," *Machine Learning Mastery*, 2016. [Online]. Available:https://machinelearningmastery.com/transform-machine-learning-data-weka/. [Accessed: 16-may-2017].

[13] A. B. Nassif, L. F. Capretz, and R. Hill, "Analyzing the Non-Functional Requirements in the Desharnais Dataset for Software Effort Estimation," *arXiv Prepr. arXiv1405.1131*, no. 2014.

[14] Z. shaukat, R. Naseem, and M. Zubair, "Software Requirement Risk Prediction Dataset." Mar-2018.. [Data set] https://doi.org/10.5281/zenodo.1209601,

[15] *Web.uvic.ca*, 2008. [Online]. Available: http://web.uvic.ca/~cloke/Seng321Designer/SENG3212008_Group4_RS1.0.doc. [Accessed: 13- Feb- 2018].

[16] *Utdallas.edu*, 2007. [Online]. Available: https://www.utdallas.edu/~chung/RE/Presentations07S/Team_1_Doc/Documents/SRS4.0.doc. [Accessed: 13- July- 2017].

[17] *Itech.fgcu.edu*. [Online]. Available: http://itech.fgcu.edu/faculty/zalewski/ism4331/his-srs.doc. [Accessed: 13- july- 2017].

[18] *Ibm.com*, 2012. [Online]. Available: https://www.ibm.com/developerworks/community/files/basic/anonymous/api/library/8afa9689-cdbb-45728a980b00e91daa77/document/12df1420-c51b-4e33-b5e1fb86d29803a3/media/STM%20srs.docx. [Accessed: 21- july- 2017].

[19] E. Kocaguneli, "cocomosdr," 2009. [Data set]. https://doi.org/10.5281/zenodo.268433. [Accessed: 09-Jan-2018].

[20] J. Desharnais, "Analyse statistique de la productivitie des projets informatique a partie de la technique des point des fonction," *Master's Thesis, University of Montreal,* 1989

[21] D. KNN, "Decision tree vs. KNN", *Datascience.stackexchange.com*, 2016. [Online]. Available: https://datascience.stackexchange.com/questions/9228/decision-tree-vs-knn. [Accessed: 10- May- 2017].

[22] L. Wallace and M. Keil, "Software Project Risks and Their Effect on Outcomes," *Commun. ACM*, vol. 47, no. 4, pp. 68–73, 2004