International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-2 Issue-4, April 2013

Software Project Health Analysis: Prediction of Outcome at Initial Stage

Deepanshu Sharma, Banwari, Deepak Upadhyay

Abstract*— The paper proposes an approach for analyzing the health of a software project. The approach aims at the prediction of software project outcome as Success or Failure at the Initial stage. The approach involves the collection of historical projects data in a defined format. The collected data is in the form of Risk Factors and their corresponding values of Impact and Probability. The collected data is then performed with some pre-processing so as to generate information (rule set) from them. The generated rule set or information can then be applied to future projects so as to predict their outcome based on the values of the Impact and Probability for existing Risk Factors. Here we have used Decision Tree Rule Induction for the generation of the rule set from the pre-processed data.

***Index Terms*— Decision Tree, Project Health Analysis, Risk Factors and Dimensions, Rule Set**

I. INTRODUCTION

A Project is a complex, non-routine, one time effort limited by time, budget, resources and performance specifications design to meet customer needs [1]. The software industry is one of the largest manufacturing industries in the world, with $350 billion in off-the-shelf software sold each year and over $100 billion in customized code on top of that. Software Risk management is a software engineering practice with processes, methods, and tools for managing risks in a project [2]. PM-BOK (Project Management Body of Knowledge) defines risks as: "an uncertain event or condition that, if it occurs, has a positive or negative effect on project's objective. On the other side, PRINCE2, the UK government sponsored project management standard defines risk as: "uncertainty of outcome" [3].

Software project risk management is a complex activity. It has to deal with uncertain events of the software projects and their causes [4]. Software project risk management is an ethic in which the project team continually asses what may negatively impact the project, determines the probability of such events occurring, and determines the impact of such events occur [5].

The main objective of project health analysis is to be able to predict whether the project is going to be completed successfully on time, within budget and that satisfies all the specified requirements. The process includes the conversion of historical project data into information and knowledge so that it can be applied to future projects to predict their outcome (success/ failure).

Manuscript published on 30 April 2013.
* Correspondence Author (s)
Deepanshu Sharma, School of Information Technology (SOIT), Center for Development of Advance Computing (C-DAC), Ministry of Communications & IT, Govt. of India, Noida, UP, India.
Banwari, School of Information Technology (SOIT), Center for Development of Advance Computing (C-DAC), Ministry of Communications & IT, Govt. of India, Noida, UP, India.
Deepak Upadhyay, School of Information Technology (SOIT), Center for Development of Advance Computing (C-DAC), Ministry of Communications & IT, Govt. of India, Noida, UP, India..

II. LITERATURE REVIEW

A. *Software Project Risk Factors and Dimensions*

There has been a lot of research on the Software Project Risk Factors that contributes in project success or failure. In an experimental study, Han and Huang [5] gave a good review on software risk research.

TABLE I
SUMMARY OF SOFTWARE RISK RESEARCH [5]

Author (Year)	Dimensions of Risks	Number of Software Risks
McFarlen (1981)	3	54
Boehm (1991)	0	10
Barki et al. (1993)	5	55
Summer (2000)	6	19
Longstaff et al. (2000)	7	32
Cule et al. (2000)	4	55
Kliem (2001)	4	38
Schmidt et al. (2001)	14	33
Houston et al. (2001)	0	29
Murti (2002)	0	12
Addision (2003)	10	28
Carney et al. (2003)	4	21

Wallace and Keil [6] proposed six dimensions of software risks that includes User, Requirements, Project Complexity, Planning and Control, Team, and Organizational Environment in 2004 in which they also specified their corresponding risk factors. Based on these literatures, we have categorized the risk factors in these six Risk Dimensions shown in Table II.

TABLE II
RISK FACTORS AND RISK DIMENSIONS

Risk Dimensions	Risk Factors
User	Lack of User Involvement. Conflict between users. Lack of user cooperation. User resistance to change. Users with negative attitude.
Requirements	Incorrect system requirements. Continuing changes for requirements. Inadequately identified requirements. Failure to manage end user expectations.
Complexity	Project involves use of new technology. Using immature technology. High level of technical complexity. Project size. Inexperience with project platform/ environment. Large number of complex interfaces.

Planning	Lack of effective project management. Inadequate estimation of resources. Unrealistic budgets. Unrealistic schedules. Project milestones not clearly defined. Not managing change properly.
Team	Lack of required skills. Conflicts among members. Unfamiliar with the tasks. Inadequately trained team members. Ineffective communication. Lack of staff commitment. Unavailability of key staff. Lack of clarity of role definitions.
Environment	Lack of senior management commitment. Unstable organizational structure. Instability in project staffing. Changing organizational management. Negative impact by corporate strategy.

B. Software Project's Classification on Outcome Basis

There has been a lot of research and surveys for the classification of software project based on their outcome. Clancy [7] from Standish Group Report 2008 presented the three categories of software projects.

i. Resolution Type 1 (project success): The project is completed on-time, on-budget and fulfilled all functions and features as specified.
ii. Resolution Type 2 (project challenged): The project is completed and operational but over-budget, over the time estimate, and offers fewer functions and features than originally specified.
iii. Resolution Type 3 (project failed): The project is cancelled at some point during the development cycle.

An analysis of the CHAOS report is presented in Table III that shows an improvement in project success based on the factors of "on-schedule, on-budget and to-specification" [8].

TABLE III
CHAOS REPORT FINDINGS [8]

Type\Year	1994	1996	1998	2000	2002
Succeeded	16%	27%	26%	28%	34%
Challenged	53%	33%	46%	49%	51%
Failed	31%	40%	28%	23%	15%

III. APPROACH FOR PROJECT HEALTH ANALYSIS: PREDICTION OF OUTCOME

The approach aims at the prediction of software project outcome as Success or Failure at the Initial stage. First step is the collection of historical projects data in a defined format. The collected data is in the form of Risk Factors and their corresponding values of Impact and Probability that are obtained in a defined range. The collected data is then performed with some pre-processing so as to generate information (rule set) from them. The data pre-processing step involves the calculation of the Entropy and Weights for each risk factor individually. These values will provide us the basis for generation of rule set that can be applied on future projects for prediction.

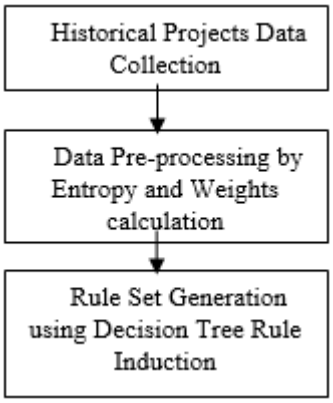

Figure 1 Framework for Analysis and Prediction

Step 1: Data Collection

The first step is the collection of historical projects data. The data is collected in a defined format as shown below in Table IV. The data is obtained through various sources that also included questionnaires. The data collected is in the form of the existing risk factors and their corresponding values of Impact and Probability for an individual project.

TABLE IV
DATA ACQUISITION TEMPLATE

Project ID	Risk Dimension	Risk Factor	Impact (1-5)	Probability (1-5)

The severity of a Risk factor can be assessed in terms of its Impact and Probability. Impact is defined as a measure of the severity of a risk's consequence if the risk were to occur. Probability (of occurrence of a risk event) is defined as a measure of the likelihood that a risk will occur. Here, in our approach, the Impact and Probability of a Risk Event is assessed on a scale of 1 to 5, where 1 and 5 represents the minimum and maximum possible impact and probability of occurrence of a risk event.

Step 2: Data Pre-processing- Calculation of Weights for each Risk Factors

The second step is the Data Pre-processing step. Xiaohong, GuoRui and Tiyun Huang suggested a framework [11]. But here, we have used a method that involves the calculation of Entropy and Weights of each risk factor. The obtained weight values for each risk factor signify the information content for the collected data and thus provide a basis for classification. Step 2 involves the following calculations.

1. Calculating Risk Exposure (V) [9], [11] for each risk factor: Risk Exposure is defined as a measure of magnitude of a risk based on the values of impact (I) and probability (P).
$$V_{factor\ f} = I_{factor\ f} * P_{factor\ f} \quad (1)$$
2. Processing Data for each risk dimension: after the calculation of risk exposure (V) for each factor, we have performed the normalization of each value in the range of 0.0 to 1.0and termed it as V'.
$$V'_{factor\ f'} = (V_{max\ (factor\ f,\ dimension\ d)} - V_{factor\ f'}) / (V_{max\ (factor\ f,\ dimension\ d)} - V_{min\ (factor\ f,\ dimension\ d)}) \quad (2)$$

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

3. Calculating Entropy (E) [10], [11] for each risk factor: Entropy is a commonly used measure in information theory.
$E_{factor\ f} = -\sum_{i=1}^{N}(V'_{factor\ f}\ log_2\ V'_{factor\ f})$,
$E_{factor\ f} \in [0,1]$. (3)
where N is the number of risk factors.
4. Calculating Weight (W) [11] for each risk factor from entropy values:
$W_{(factor\ f,\ dimension\ d)} = (1 - E_{factor\ f}) / (N_{dimension\ d} - \sum_{i=1}^{N} E_{dimension\ d})$ (4)

where N is the number of risk factors in a risk dimension d. After performing all these computations, we obtained a specific value of weight for each risk factor. These values will provide a basis for information generation.

Step 3: Rule Set Generation using Decision Tree Induction

Decision Trees are considered to be one of the most popular and powerful tools for classification and prediction. Decision Tree induction process automatically construct a decision tree from a given dataset [12] [13]. The tree structure has following nodes and is characterised as:

1. Root node: risk dimension as the root node that has no incoming node and contains the information of its risk factors and their corresponding weight values.
2. Leaf Node: is the decision node that indicates the value of target attribute known as label.

An attribute is decided to be at the leaf node that we wish to predict as the output attribute that is the values either Success or Failure.

IV. EXPERIMENTS AND RESULTS

For the generation of information or rule set from collected and processed data, the decision tree rule induction technique produced the following results.

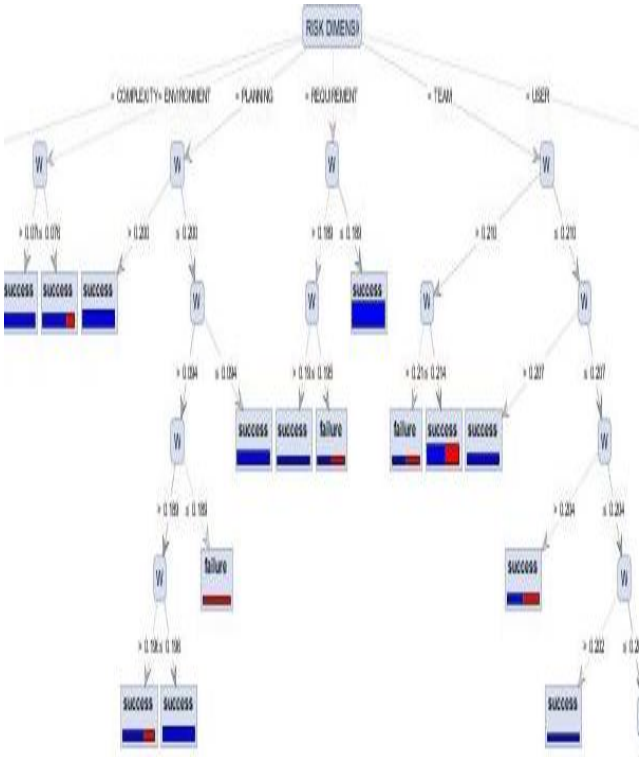

Figure 2 Generated Decision Tree

The text view of the obtained rule is as follows:

```
RISK DIMENSION = PLANNING
|   W > 0.200: success {success=6, failure=0}
|   W ≤ 0.200
|   |   W > 0.094
|   |   |   W > 0.189
|   |   |   |   W > 0.196: success {success=2, failure=1}
|   |   |   |   W ≤ 0.196: success {success=5, failure=0}
|   |   |   W ≤ 0.189: failure {success=0, failure=1}
|   |   W ≤ 0.094: success {success=4, failure=0}
RISK DIMENSION = REQUIREMENT
|   W > 0.189
|   |   W > 0.195: success {success=2, failure=0}
|   |   W ≤ 0.195: failure {success=1, failure=1}
|   W ≤ 0.189: success {success=9, failure=0}
RISK DIMENSION = TEAM
|   W > 0.210
|   |   W > 0.214: failure {success=1, failure=1}
|   |   W ≤ 0.214: success {success=4, failure=3}
|   W ≤ 0.210
|   |   W > 0.207: success {success=3, failure=0}
|   |   W ≤ 0.207
|   |   |   W > 0.204: success {success=1, failure=1}
|   |   |   W ≤ 0.204
|   |   |   |   W > 0.202: success {success=1, failure=0}
|   |   |   |   W ≤ 0.202
|   |   |   |   |   W > 0.101
|   |   |   |   |   |   W > 0.201: success {success=2, fa
|   |   |   |   |   |   W ≤ 0.201: success {success=1, fa
|   |   |   |   |   W ≤ 0.101: success {success=2, failur
RISK DIMENSION = USER
```

Figure 3 Text View of Obtained Rules

The decision tree is generated with Information Gain as the splitting criteria and at a confidence of 0.25.

V. CONCLUSION

The paper includes a method for software project health analysis for prediction of project outcome (success/failure) at initial stage using Decision Tree rule induction technique. Using this method for new project collected data at initial stage and the obtained rules, the project stakeholders can predict the outcome and thus take the respective actions to mitigate risks.

VI. APPENDIX

A sample format of the questions in questionnaires is included here used for the data collection purpose apart from other resources.

TABLE V
SAMPLE QUESTIONS

QID	Questions
D1.1	What was level of involvement of user?
D2.2	Are the requirements continually changing?
D6.2	Is the organizational structure unstable?
D4.3	Are the estimation of budgets and schedules realistic?
D5.6	Is there any lack of staff commitment?
D5.1	Is there any lack of required skills among the key staff?
D3.1	Does the project involve a high level of technical complexity?

VII. ACKNOWLEDGMENT

We would like to thank our Professors for providing us with proper support and guidance. This work is the result of their valuable contribution that they gave us throughout the research.

REFERENCES

[1] Gray, C. F. and Larsen, E.W., "*Project Management: The Managerial Process*", 4th edition, McGraw-Hill Educations, Singapore, 2008.
[2] Abdullah Al Murad Chowdhury and Shamsul Arefeen, "*Software Risk Management: Importance and Practices*", IJCIT vol. 02, issue 01, 2011.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

[3] B. Hughes and M. Cotterell, Software Project Management 4th edition, pp.147, 1996, McGraw Hill (UK).
[4] Tharwon Arnuphaptrairong, "*Top Ten Lists of Software Project Risks: Evidence from Literature Survey*", IMECS vol. 1, Hong Kong, 2011.
[5] W.M. Han and S.J. Huang, "*An Empirical analysis of Risk Components and Performance of Software Projects*", The Journal of Systems and Software vol. 80 number 1, pp.42-50, 2007.
[6] L. Wallace and M. Keil, "*Software Project Risk and their Effect on Outcomes*", Communication of the ACM, vol. 47 number 4, pp. 68-73, 2004.
[7] Clancy, T. *The Standish Group Report*, Retrieved Feb 20, 2008 from http://www.projectsmart.co.uk/reports.html, Chaos report, 1995.
[8] Wasileski, J.S., "*Learning Organization Principles & Project Management",* SIGUCCS'05, November 6-9, 2005, Monterey, California, USA.
[9] Salvatore Alessandro Sarcia, Giovanni Cantone and Victor R. Basili, "*A Statistical Neural Network Framework For Risk Management Process*", from the proposal to its Preliminary Validation for Efficiency.
[10] Linyu Yang, Dwi H. Widyantoro, Thomas Ioerger, John Yen, "*An Entropy based Adaptive Genetic Algorithm for Learning Classification Rules*".
[11] Xiaohong Shan, GuoRui Jiang and Tiyan Huang, "*A Framework of estimating software project success potential based on association rule mining*", IEEE, 2009.
[12] Lior Rokach and Oded Maimon, "*Decision Trees*", Chapter 9, pp. 165-192.
[13] Ding-An Chiang, Wei Chen, Yi-Fan Wang and Lain-Jinn Hwang, "Rules Generation from the Decision Tree", Short paper, Journal of Information Science and Engineering, pp. 325-339, 2001.

Deepanshu Sharma received his Bachelor of Technology (B.Tech) degree in Information Technology in 2011 from Mahatma Gandhi Mission's College of Engineering & Technology, Noida, U.P. (India). He is currently working towards Master of Technology (M.Tech) degree in Information Technology at School of Information Technology (SOIT) from Center for Development of Advance Computing (C-DAC), Noida, U.P. (India) under the Ministry of Communications and Information Technology, Government of India. His area of research includes Software Engineering and Computer Network's Management and Security.

Banwari received his Bachelor of Technology (B.Tech) degree in Computer Science and engineering in 2010 from Maharaja Agrasen Institute of Technology (MAIT) Delhi, India. He is currently working towards Master of Technology (M.Tech) degree in Information Technology at School of Information Technology (SOIT) from Center for Development of Advance Computing (C-DAC), Noida, U.P. (India) under the Ministry of Communications and Information Technology, Government of India. His area of research includes Software Engineering, Computer Network's Management and Security, Algorithms, Mobile computing and Artificial intelligence.

Deepak Upadhyay received his Bachelor of Technology (B.Tech) degree in Information Technology in 2011 from Govt. Engineering college Ajmer Rajasthan, India. He is currently working towards Master of Technology (M.Tech) degree in Information Technology at School of Information Technology (SOIT) from Center for Development of Advance Computing (C-DAC), Noida, U.P. (India) under the Ministry of Communications and Information Technology, Government of India. His area of research includes GSM, Computer Network, and algorithms.