RESEARCH ARTICLE OPEN ACCESS

Selection of Software Requirements using Decision Tree during Software Development Process

Syed Zeeshan Hussain*, Mohd. Shahid**, Chaudhary Wali Mohammad***
**Department of Computer Science, Faculty of Natural Sciences, Jamia Millia Islamia (A Central University), New Delhi-110025, INDIA*
***Ph.D Research Scholar, Computer Science and Technology Research Group, Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia (A Central University), New Delhi-110025, INDIA*
****Computer Science and Technology Research Group, Department of Applied Sciences and Humanities, Faculty of Engineering and Technology, Jamia Millia Islamia (A Central University), New Delhi-110025, INDIA*
Corresponding Author: Syed Zeeshan Hussain

ABSTRACT
Tree is an important data structure which is used in software development process (SDP). Different types of tree are used in literature like binary search tree, heap tree, decision tree etc; in SDP. In this paper, we present for the selection of software requirements on the basis of cost in SDP. Finally, the utilization of the proposed work is demonstrated with the help of a case study, which based on Institute Examination System.
Keywords: Decision Tree, AND/OR Tree, Function Point

Date of Submission: 11-09-2017 Date of acceptance: 28-09-2017

I. INTRODUCTION

Data structure is an important attribute of a program [1]. It is a logical and mathematical organization of data. Data structure is mainly divided into two parts, i.e., "*linear data structure and non-linear data structure*". There are different applications of the linear and non-linear data structure in the area of software engineering. For example, in requirements elicitation process, AND/OR Graph or AND/OR Tree is used to elicit and model the software requirements [2, 3]. In our work, we have identified different applications of the non-linear data structure in software engineering. For example, in [4], we proposed a "*fuzzy attributed goal oriented software requirements analysis method*", called FAGOSRA. In another study, we proposed a "*graph theory based algorithm for the computation of Cyclomatic complexity of software requirements*" [5]. There are some applications of the tree in the area of computer science and technology (CST) like decision tree (DT), etc. Therefore, in this work, we mainly focus on the DT and its application in software engineering.

A decision tree (DT) uses the tree like model to illustrate every possible outcome of a decision. In DT, each branch represents a possible decisions, occurrences or reaction. There are different applications of the DT in the area of CST, for example, in data mining, DT are used to classify the different types of the objects. In 2013, Franco-Arcega [7], discussed the application of the DT for the classification of astronomical objects like galaxies and stars. DTs have also been applied for the selection problem. For example, Harwati and Sudiya [6] discussed the application of DT for the selection of student's model.

Software engineering is an important research area of CST in which we mainly deal with the software designing, development, maintenance, and deployment of the software product [1]. DTs have been successfully applied in the area of CST; and in this paper we discuss the application of the DT for the selection of the software requirements on the basis of the cost of each requirement of the software. In the literature of the software engineering, different methods have been developed for the selection of software requirements (SR). For example, Sadiq and Jain [3] proposed a "*fuzzy based method for the selection of goals in goal oriented requirements elicitation process*". Based on our literature review, we identify that little attention is given for the selection of SR using DT. Therefore, in this paper, we proposed a method for the selection of software requirements using DT.

The remaining part of the paper is organized as follows: Related work in the area of software requirements selection is given in section 2. In section 3, we present the proposed method for the

selection of SR using DT. Case study based on proposed method is given in section 4. Finally, conclusions and future work are given in section 5.

II. RELATED WORK

In this section, we discuss the related work in the area of the selection of software requirements (SR). In SR selection problem, different requirements are given and the objective is to select the requirements according to the need of the stakeholders. SR selection problem is also referred to as "Next Release Problem" in search base software engineering [15]. Different methods have been developed for the selection of SR. For example, Li [8] proposed a "*multi-objective optimization technique for the selection of the SR on the basis of the cost, revenue, and uncertainty*". Cheng et al. [9] proposed an "*adaptive memetic algorithm based on "Multi-objective optimization for software next release problem*". Veerapen *et al.* [10] proposed "an integer linear programming approach to the single and bi-objective next release problem". Teaching learning based optimization method was used by the Chaves-Gonzalez et al. [11] for the SR selection. A swarm intelligence evolutionary algorithm was used by the Chaves-Gonzalez [12] for the optimization of the SR. Del Sagrado et al. [13] used the "*multi-objective ant colony optimization for requirements selection*". In another study, Chaves-Gonzalez and Perez-Toledano [14] apply the differential evolution with pareto-tournament for the multi-objective next release problem. A robust "*optimization approach to the next release problem in the presence of uncertainties*" was proposed by Paixao and Souza [16]. Above methods are based on "*search based software engineering*" (SBSE) [15].

There are some other studies, which are based on non-SBSE. These studies are mainly based on the "*multi-criteria decision making*" (MCDM) methods. For example, Fernandes *et al*. [17] compare the two methods for prioritizing the software requirements, i.e., AHP i.e., "*analytic hierarchy process*" and ELECTRE I. In another study, Sadiq *et al*. [18] used the AHP "*for the selection and prioritization of the software requirements*" using AHP method. Based on our analysis, we identify the following issues which are present in the literature of application of non-linear data structure to software engineering:

- Research Issue-1: How to apply the decision trees (DT) during the selection of the software requirements?
- Research Issue -2: How to evaluate the DT on the basis of different criteria like cost, performance, reliability, etc.? Here, we will focus on the cost only.

To address the above issues, we proposed a method for the selection of SR using DT when cost is used as the criteria for the selection of SR. The detailed description of the proposed method is given in the next section.

III. PROPOSED METHOD

In this section, we proposed a method for the selection of SR using DT when cost is used as a criterion. Proposed method includes the following steps:

Step 1: Identify and prioritize the stakeholders
Step 2: Compute the cost of each requirement
Step 3: Construct the decision tree
Step 4: Select the software requirements on the basis of cost

Step 1: Identify and prioritize the stakeholders

A stakeholder can be defined as "any group or individual who can affect or is affected by the achievement of the organization". Stakeholder identification is an important activity before the development of the software projects. In any project, several stakeholders are involved for the development of the successful software product; and it is not possible to consider all the stakeholders during the development process. Therefore, the objective of this step is to identify and prioritize the stakeholders before the identification of the software requirements. Here, we apply the steps proposed by Sadiq [19] for the identification and prioritization of the stackholders. In [19] following steps are given: (a) "*specify stakeholder types and their roles*" (b) "*select and classify requirements*"and (c) "*stakeholders analysis*".

Step 2: Compute the cost of each requirement

Cost is an important criterion for the selection of the software requirements. Based on our review, we identify that in literature little attention is given to the computation of the cost of each requirement. Therefore, in this method, we practically compute the cost of each requirement by using the function point (FP) approach. FP is a software metric, which is used to describe the functionality of software. The computation of the FP depends on the five different parameters like "external input (EI)", "external output (EO)", "external query (EQ)", "internal logical file (ILF)", and " external interface file (EIF)"; it also depends on the 14 general system characteristics [1]. In India, the cost of the implementation of one FP is $125 (approximately) [20].

Step 3: Construct the decision tree

In this step, we construct the decision tree on the basis of the requirements of software. In this tree, each node contains the requirements along with its cost.

Step 4: Select the software requirements on the basis of cost

The objective of this step is to select the software requirements on the basis of the cost.

IV. CASE STUDY

The objective of Institute Examination System (IES) is to deal with the examination related activities like submission of the examination form, submission of the examination fee, generation of the hall ticket, generation of the mark-sheet, etc. The explanation of the proposed method is given below:

Step 1: In IES, we have identified the following stakeholders(s), who will participate during the development of the software product: (i) Financers (S1), (ii) Director of the Institute/VC of the University (S2), (iii) Requirements analysts for the functional requirements of the IES (S3), (iv) Requirements analyst for the non-functional requirements of the IES (S4), (v) Developer (S5), and (vi) Tester (S6). In this step, "*we prioritize the stakeholders on the basis of the importance of the functional requirements*". On the basis of the results of [19], we identify that S1 and S2 have highest priority. Stakeholder S4 has the second priority and stakeholder S3 has the third priority. Stakeholder S5 and S6 have the fourth priority.

Step 2: In this step, we compute the cost of each requirement of IES.

R1: Login Module

For the computation of the cost of the software requirements, we first visualize the requirements in the same as it would be displayed on the computer screen after implementation. Such type of visualization helps to find out the following parameters which are used in the computation of the function point, i.e., EI, EO, EQ, ILF, and EIF. In our case study, we first compute the cost of the login module of the IES. The login module of the IES is exhibited in Fig. 1.

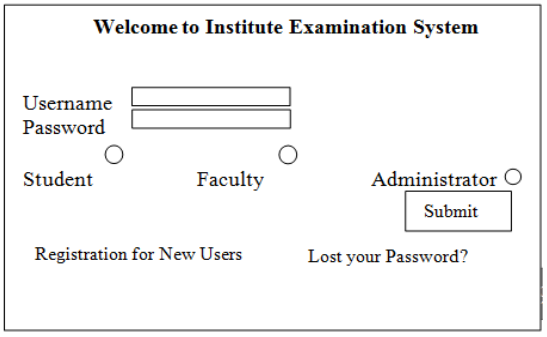

Fig. 1: Login Part of IES

In Fig. 1, there are 8 EI, i.e., EI-1: Username, EI-2: Password, Selection of the type of user, for example, EI-3: Student, EI-4: Faculty and EI-5: Administrator, EI-6: Submit button, and EI-7: Lost your password, and the EI-8: registration for new users. There would be three different EO depending on the selection of the type of the users, i.e., Student, Faculty, and Administrator. There are two EQ, i.e., at the time of the "Lost you password", the query would be in terms of "Yes" or "No"; and another query would be at the time of the "registration for new users". There is only one single ILF, and in our case study, there is no EIF. Finally, we have the following values to compute the unadjusted function point (UFP), when the complexity of the IES is assumed to be average:

EI = 8; EO= 3; EQ = 2; ILF = 1; and EIF = 0
The computation of the UFP is given in Table 1.

Table 1: Computation of UFP for login module

S. No.	Functional Units	Complexity (Average)	Functional Unit
1	EI =8	4	32
2	EO = 3	5	15
3	EQ = 2	4	8
4	ILF = 1	10	10
5	EIF = 0	7	0
Total UFP			65

Now we compute the complexity adjustment factor by determining the values of the 14 general system characteristics (GSC). The list of the GSC is given below:

GSC-1: "Does the system require reliable backup and recovery?
GSC-2: Is data communication required?
GSC-3: Are there distributed processing functions?
GSC-4: Is performance critical?
GSC-5: Will the system run in an existing heavily utilized operational environment?
GSC-6: Does the system require online data entry?
GSC-7: Does the online data entry require the input transactions to be built over multiple screens or operations?
GSC-8: Are the master file updated online?
GSC-9: Is the inputs, outputs, or inquiries complex?
GSC-10: Is the internal processing complex?
GSC-11: Is the code designed to be reusable?
GSC-12: Are conversion and installation included in the design?
GSC-13: Is the system designed for multiple installations in different organizations?
GSC-14: Is the application designed to facilitate change and ease of use by the user?"

Now we evaluate 14 GSC on the scale of 0 to 5. The meaning of the scale is given below:

- 0: No Influence
- 1: Incidental
- 2: Moderate
- 3: Average
- 4: Significant
- 5: Essential

The results after evaluation on the basis of the above scale are given in Table 2. Here, we assume that all GSC are significant.

The complexity adjustment factor (CAF) would be calculated as: $(0.65+0.01X \sum GSC_i)$

CAF = 0.65+0.01X64 = 1.29

Finally, the value of the FP can be calculated as:

FP = UFP X CAF = 65X1.29 = 83.85

The cost of the login module according to the Indian Software Industry [20] would be $125X83.85 = $ 10, 481.

Table 2: Evaluation of 14 GSC on the scale of 0 to 5

S. No.	GSC	Values
1	GSC_1	4
2	GSC_2	4
3	GSC_3	4
4	GSC_4	4
5	GSC_5	4
6	GSC_6	4
7	GSC_7	4
8	GSC_8	4
9	GSC_9	4
10	GSC_{10}	4
11	GSC_{11}	4
12	GSC_{12}	4
13	GSC_{13}	4
14	GSC_{14}	4
Total		64

R2: Submission of Examination Form

R2: For submission of exam form, there are 15 EI, i.e., EI-1:Students's Name, EI-2: Father's Name, EI-3: Date of Birth, EI-4: Enrolment No, EI-5: Nationality, EI-6:Category, EI-7: Medium of Examination, EI-8: Residential Address, EI-9: Permanent Address, EI-10: Details of Exam Passed, EI-11:Paper code for which appearing, EI-12:Practical (if any), EI-13: Photo Upload, EI-14: Signature Upload. There would be one EO i.e. Fee Submission. There are only one EQ i.e "Have you submitted the College fee", the answer would be in terms of "Yes" or "No"; if "yes" then hall Ticket will generate and if "No" then it will redirected to Payment Gateway. There is only one ILF and two EIF, i.e., EIF-1: Payment of Exam fees, EIF-2: Aadhar Card. The computation of the UFP is given in Table 3.

Table 3: Computation of UFP for Submission of Examination Form

S. No.	Functional Units	Complexity (Average)	Functional Unit
1	EI =14	4	56
2	EO = 1	5	5
3	EQ = 2	4	8
4	ILF = 1	10	10
5	EIF = 2	7	14
Total UFP			93

The complexity adjustment factor (CAF) would be calculated as:
CAF = 0.65+0.01X64 = 1.29

Finally, the value of the FP can be calculated as:

FP = UFP X CAF = 93X1.29 = 119.97

The cost of the Submission of Examination From according to the Indian Software Industry [20] would be $125X119.97 = $ 14996.25.

R3: Submission of Exam Fees

For submission of Exam Fee, there are 7 EI, i.e., EI-1: Students's Name, EI-2: Branch Name, EI-3: Year/Semester, EI-4: Enrolment No, EI-5:Total Amount, EI-6 Submit, EI-7: Print. There would be one EO i.e. E-Receipt of Exam fees, one ILF and one EIF i.e. Payment Gateway. There is no EQ. The computation for R3 is given in table 4.

Table 4: Computation of UFP for Generation of Hall Ticket

S. No.	Functional units	Complexity (Average)	Functional Unit
1	EI =7	4	28
2	EO = 1	5	5
3	EQ = 0	4	0
4	ILF = 1	10	10
5	EIF = 1	7	7
Total UFP			50

The complexity adjustment factor (CAF) would be calculated as:
CAF = 0.65+0.01X64 = 1.29

Finally, the value of the FP can be calculated as:

FP = UFP X CAF = 50X1.29 = 64.50

The cost of the Submission of Examination From according to the Indian Software Industry [20] would be $125 X 40.56 = $ 8062.5

R4: Generation of Hall Ticket

For the generation of Hall Ticket, there are 6 EI, i.e., EI-1:Students'id or Email Id, EI-2: Password, EI-3: Captcha, EI-4: Forget Password, EI-5: Change Password, EI-6 Submit, There would be one EO i.e E-Version of Hall Ticket. There is only one ILF .There is no EQ and EIF. The computation of UFP for R4 is given in Table 5.

The complexity adjustment factor (CAF) would be calculated as:
CAF = 0.65+0.01X64 = 1.29

Table 5: Computation of UFP for Generation of Hall Ticket

S. No.	Functional Units	Complexity (Average)	Functional Unit
1	EI =6	4	24
2	EO = 1	5	5
3	EQ = 0	4	0
4	ILF = 1	10	10
5	EIF = 0	7	0
Total UFP			39

Finally, the value of the FP can be calculated as:

FP = UFP X CAF = 39X1.29 = 50.31

The cost of the Submission of Examination From according to the Indian Software Industry [20] would be $125 X 40.56 = $ 6288.75

R5: Generation of Mark-sheet

For generation of Result, there are 8 EI, i.e., EI-1: Login, EI-2: Branch, EI-3: Year, EI-4: Semester, EI-5 Submit, EI-6: click here to download Mark-sheet, EI-7: Print.EI-8: Logout. There would be one EO i.e. E-Version of Result. There is only one ILF .There is no EQ and EIF. The computation of UFP for R5 is given in Table 6.

The complexity adjustment factor (CAF) would be calculated as:

CAF = 0.65+0.01X64 = 1.29

Finally, the value of the FP can be calculated as:

FP = UFP X CAF = 47X1.29 = 60.30

Table 6: Computation of UFP for Generation of Hall Ticket

S. No.	Functional Units	Complexity (Average)	Functional Unit
1	EI =8	4	32
2	EO = 1	5	5
3	EQ = 0	4	0
4	ILF = 1	10	10
5	EIF = 0	7	0
Total UFP			47

The cost of the Submission of Examination From according to the Indian Software Industry [20] would be $125 X 40.56 = $ 7578.75

Step 3: Construct the decision tree

In this step, we construct the decision tree for different requirements of R1, R2, R3, R4, and R5. The decision tree of our work is exhibited in fig 2. In this tree the high level objective, i.e. IES, is refined and decomposed into 5 requirements. In this tree the cost of each requirements is attached with every node of the tree. Here, the nodes are in requirements of the software engineering.

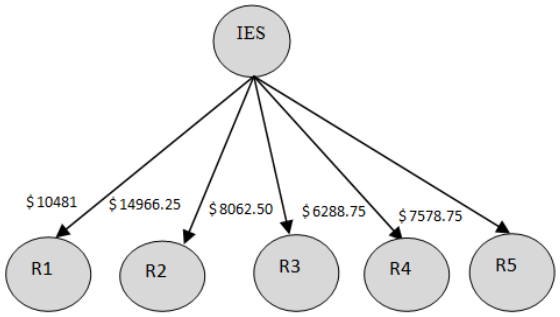

Fig 2: Decision Tree

R1: Login Module
R2: Submission of Examination form
R3: Submission of Examination Fees
R4: Generation of Hall Ticket
R5: Generation of Mark-Sheet

Similarly, we have calculated the cost of the remaining modules; and the results are shown in Table7.

Step 4: Select the requirement on the basis of cost

The objective of this step is to select the requirements from the DT, as shown in fig. 2. Such type of DT would be used to decide which requirements would be implemented depending on the cost of requirements. In our example, first login module would be implemented and the cost of this requirement id $10,481. Among different requirements, those requirements would be implemented which are within budget.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a method for the selection of software requirement. Proposed method includes the following steps: (i) identify and prioritize the stakeholders, (ii) compute the cost of each requirement, (iii) construct the decision tree, and (iv) select the software requirements on the basis of cost. we have applied proposed method on Institute Examination System (IES). In our work we have identified following requirements of IES i.e., (a) login Module, (b) submission of examination form, (c) Submission of exam fee (d) generation of hall ticket, and (e) generation of mark sheet. After applying the proposed method on these requirements we identify that every requirement has its own cost for example the cost of login module is $ 10481 according to Indian software industries. In our proposed method we have used decision treein order to decide which requirement would be implemented during software development process. Future research includes the following:

(1) To select the software requirement using decision tree when more than one criterion is used.
(2) To develop a fuzzy based method for the selection of software requirements. In which decision tree would be used for decision making purpose.

REFERENCES

[1] Pressman R. S., Software Engineering: A Practitioner's Approach, 5th Edition, McGraw-Hill International Edition, Computer Science Series. 2001.
[2] Sadiq M and Jain S K., "Applying Fuzzy Preference Relation for Requirements prioritization in Goal Oriented Requirements Elicitation Process", International Journal of System Assurance Engineering and Maintenance, Springer, Vol. 5, Issue. 4, pp. 711-723, 2014.
[3] Sadiq M. and Jain S. K., "A Fuzzy Based Approach for the Selection of Goals in Goal Oriented Requirements Elicitation Process", International Journal of System Assurance Engineering and Maintenance, Springer, Vol. 6, Issue 2, pp. 157-164, June 2015
[4] Mohammad C. W., Shahid M., and Husain S. Z., " FAGOSRA: Fuzzy Attributed Goal Oriented Software Requirements Analysis Method", 9th International Conference on Contemporary Computing, Organized by Jaypee Institute of Information Technology, Noida, India, pp. 384- 389, August, 2016.
[5] Mohammad C. W., Shahid M., and Husain S. Z., "AGraph Theory based Algorithm for the Computation of Cyclomatic Complexity of Software Requirements", ICCCA-2017.
[6] Harwati and Sudiya A., "Application of Decision Tree Approach to Student Selection Model-A Case Study", In: IOP Conference Series: Materials Science and Engineering, 2016, Vol. 105.
[7] Franco-Arcega A., Flores-Flores L. G., Gabbasov R. F., "Application of Decision Tree for Classifying Astronomical Objects", 12th Mexican International Conference on Artificial Intelligence-2013, pp. 181-186.
[8] Li L., Harman M., Letier E., Zhang Y., "Robust Next Release Problem: Handling Uncertainty during Optimization", GECCO-2014, pp. 1247-1254.
[9] Cheng X., Huang Y., Cai X., Wei O., "An Adaptive memetic Algorithm based on Multi-objective Optimization for Software Next Release Problem", GEECO-2014, pp. 185-186.
[10] Veerapen N., Ochoa G., Harman M., and Burke E., "An Integer Linear Programming approach to the Single and bi-objective Next Release Problem", Information and Software Technology, vol. 65, pp. 1-13, 2015.
[11] Chaves-Gonzalez J. M., Perez-Toledano M. A., and Navasa A., "Teaching Learning based Optimization with Pareto Tournament for the Multiobjective Software Requirements Selection", Engineering Application of Artificial Intelligence, Vol. 43, pp. 89-101, 2015.
[12] Chaves-Gonzalez J. M., Perez-Toledano M. A., and Navasa A., " Software Requirements Optimizations using a multiobjective swarm intelligence evolutionary algorithm, knowledge based system, 83(2015), pp 105-115.
[13] Del Sagrado J., Del Aguila I. M., and Orellana F. J., "Multiobjective Ant Colony Optimization for Requirements Selection", Empirical Software Engineering, Vol. 20, No. 3, pp. 577-610, 2015.
[14] Chaves-Gonzalez J.M. and Perez-Toledano M. A., "Differential Evolution with Pareto Tournament for the Multiobjective Next Release Problem", Applied Mathematics and Computation, Vol. 252, pp. 1-13, 2015.
[15] Pitangueira A. M., Maciel R. S., and Barros M., "Software Requirements Selection and Prioritization using SBSE Approaches: A Systematic Review and Mapping of the Literature", Journal of Systems and Software, Vol. 103, pp. 267-280, 2015.
[16] Paixao M. and Souza J., "A Robust Optimization Approach to the Next Release Problem in the Presence of Uncertainties",

Journal of Systems and Software, Vol. 103, pp. 281-295, 2015.

[17] Fernandes J. M., Rodrigues S. P., and Casta L. A., "Comparing AHP and ELECTRE I for prioritizing Software Requirements", IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel Distributed Computing, 2015, pp. 1-8.

[18] Sadiq M., Ahmed J., Asim M., Qureshi A., and Suman R., "More on Elicitation of Software Requirements and Prioritization using AHP", IEEE International Conference on Data Storage and Data Engineering, 2010, pp. 230-234.

[19] Czarnacka-Chorobot B., " What is the cost of ONE IFPUG Method function point", World Computing proceeding SER2400, 2012.

[20] Sadiq M., "A Fuzzy- Set based Approach for the Prioritization of the Stakeholders on the basis of the Importance of Software Requirements", IETE Journal of Research, 2017.

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

Syed Zeeshan Hussain. "Selection of Software Requirements using Decision Tree during Software Development Process." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 9, 2017, pp. 71–77.