iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** XI **Month of publication:** November 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 4 Issue XI, November 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Recent Development in Software Project Evaluation Techniques

Prof Manjula R[1], Vishnu Vardhan Reddy[2], Prem Sai[3]
Computer Science Department (SCOPE), VIT University

Abstract— The main outline of this paper is to provide an idea to study various techniques of software evaluation and analyse them in an efficient and effective manner. We will actually see how various methodologies have been adopted over recent years to counter the problem faced based on risk factor, intangible factors. Here we can see how the risk management techniques are employed to avoid risks. We will see how the qualitative weighing is better than numerical weighted lists and how the operational costs have been reduced by adopting Commercial Off-The Shelf COTS) products of software.
Keywords— Risk factor, operational costs, objectives, qualitative weighing, software evaluation.

I. INTRODUCTION

Whenever a buy analysis is performed we have to select a product which satisfies all the primary requirements and costs. The evaluation of necessary requirements should be done in an effective and efficient manner. For this purpose decision analysis spread sheet can be a proper tool for the evaluation purpose. In spite of huge research going on how to tackle the software projects still there are flaws in achieving milestones within the given time frame and budget. Most of the failures occur due to the project managers not being pro-active and trying to reach out for the situation when it happens and correcting at that particular moment. A good management of the risks related to the project will always be assured as a solution to the ongoing problems. Evaluation criteria of software projects has become much more important now a days because of the rapid development in software field with more number of software projects of which only few are good products. Secondly certain standards should be met while developing these projects. Keeping in mind of these standards, the European Academic Software Award (EASA) was created. In modern days evaluation of candidate COTS based systems has to be done at the initial stage of the development process. The success rate of the evaluation is low because these evaluations are based on the initial requirements while the requirements may change over the period of time. To avoid this problem COTS software evaluation has to be done simultaneously with respect to the requirements reducing the cost, time and complexity. Although there is steady growth in software development methodologies but also there is a need for choosing an appropriate methodology and also how it effects the surrounding environment. Software testing is performed to check and analyse how well is a particular software working. It basically detects errors and defects. It prevents errors and clarifies the concerns related to system specifications. It improves quality of the particular software. It verifies and validates the system behaviour. Under prevention ways to prevent and reduce the errors. The main objective here is to develop a standard process for evaluating various software development procedures.

II. QUANTITATIVE METHODS FOR SOFTWARE DEVELOPMENT EVALUATION

A. Decision Analysis Spreadsheet

Basically a decision analysis spreadsheet helps any business organization in way that it shows the comparative analysis of different products by developing a selection criteria and also distributing weights to the selected criteria. Among all the criteria the best one is selected which gives the best score. Based on this evaluation the best product is selected.
Let us consider an example to understand the method where two products P1 and P2 are compared based on various criteria named as Items. Each of these items have their respective weights. While conducting the evaluation all the items are assigned with a raw value which gives the weighed score. Then the weighted scores for both the products P1 and P2 are compared. The key for this method is total weights for all the criteria should be equal to 100%.

		Software Alternatives				
		System 1			System 2	
Item	Decision Criterion	Weight	Raw	Weighted	Raw	Weighted
A	Presentation based on rules	20%	1.0	20.00%	1.0	20.00%

Volume 4 Issue XI, November 2016
ISSN: 2321-9653

B	Reliable	10%	1.0	10.00%	1.0	10.00%
C	Accessible	10%	1.0	10.00%	1.0	10.00%
D	Seller maturity	10%	0.5	5.00%	1.0	10.00%
E	Seller support	10%	0.5	5.00%	1.0	10.00%
F	Cost of ownership	10%	0.0	0.00%	1.0	10.00%
G	Extensible	10%	1.0	10.00%	1.0	10.00%
H	Vendor solution	5%	-0.5	-2.50%	1.0	5.00%
I	Visual rules administration	15%	1.0	15.00%	0.5	7.50%
Total		100%		72.5%		92.50%

There is another aspect of this method where we have a subgroup for a particular item. Each item is further classified into another subgroup where the sub-weights add up to 100%. The total score for this particular criteria or item is calculated by applying multiplication operation between the weight of the item and total score of the subgroup. This method is illustrated in the example shown below.

		Software Alternatives				
			System 1		System 2	
Item	Decision Criterion	Weight	Raw	Weighted	Raw	Weighted
A	Graphical user interface	20%		18.00%		11.50%
A.1	Multiple window use	50%	1.0	50%	0.5	25%
A.2	Resizable windows	30%	1.0	30%	1.0	30%
A.3	Remembers user's screen settings	10%	0.5	5%	-0.5	-3%
A.4	Provides keyboard shortcuts	10%	0.5	5%	1.0	5%
Subtotal			90%		58%	

The description of raw value is as follows:

Score Value	Definition
1.0	Item fully decision criterion.
0.5	Item partially satisfies decision criterion.
0.0	Null (The item neither satisfies nor dissatisfies decision criterion)
-0.5	Item partially dissatisfies decision criterion.
-1.0	Item fully dissatisfies business requirement or decision criterion.

B. Empirical Strategy

In this method we use correlation and regression analysis model on Durbin-Watson Statistic and analyse the risks. The formulae for the methods are as follows:

1) Correlation Analysis: Correlation between risk factors and control factors is determined by Correlation Coefficient(r):

$$r=\frac{n\left[\sum\left(x_i,y_i\right)\right]-\left(\sum x_i\right)\left(\sum y_i\right)}{\sqrt{\left[n\left(\sum x_i^2\right)-\left(\sum x_i\right)^2\right]\left[n\left(\sum y_i^2\right)-\left(\sum y_i\right)^2\right]}}$$

2) Regression Analysis Model:

Y = b0 + b1X1 + b2X2 + . . . + bnXn where b0, b1….bn are regression coefficients and X1, X2…Xn are independent variables and Y is a dependent variable

3) Durban Watson Statistic (D):

$$\mathrm{D}=\frac{\sum\left(e_i-e_{i-1}\right)^2}{\sum e_i^2}$$

e (i) is the errors associated with observation at time t.

III. QUALITATIVE METHODS FOR SOFTWARE DEVELOPMENT EVALUATION

A. Qualitative Weight and Sum (QWS)

Step1: Performing the analysis for various criteria.

In this step all the criteria are listed and then accordingly weighted. No numeric operation are performed. But numeric operations are converted into symbols which does the required operations.

Step2: Criteria Weighting

Step2.1: Elimination the evaluands

The evaluands which does not satisfy the criteria are eliminated. Also evaluands should not take more time. In such case also the evaluands are eliminated which means the response time of evaluands should be minimum.

Step2.2: Elimination of evaluands with 0-criteria

The evaluands which are irrelevant can be eliminated. The remaining evaluands can be evaluated and can be used for future analysis.

Step2.3: Elimination of evaluands with uniform criteria

The evaluands with uniform criteria can be eliminated to avoid redundancy. By doing this we can decrease the response time of the operations.

Step3: Ranking

Now we will count number of *, # and + for the evaluands declared at the beginning. Depending on th3e count we will give rankings and the path is determined.

IV. EVALUATION TECHNIQUES FOR COTS BASED SOFTWARE

The traditional approach for testing software products are Black box testing and White box testing. With the help of COTS based systems we can have constraints which are distinctive in nature and we can perform effective testing. Basically in white box testing the process of testing is performed typically at source code level. One of the method is basis-path testing. In this testing various independent paths are analysed with the help of a code module. This type of testing is basically done when the software product is being developed and simultaneously the code is generated. It verifies whether the system satisfies all the required functionalities are implemented in correct manner.

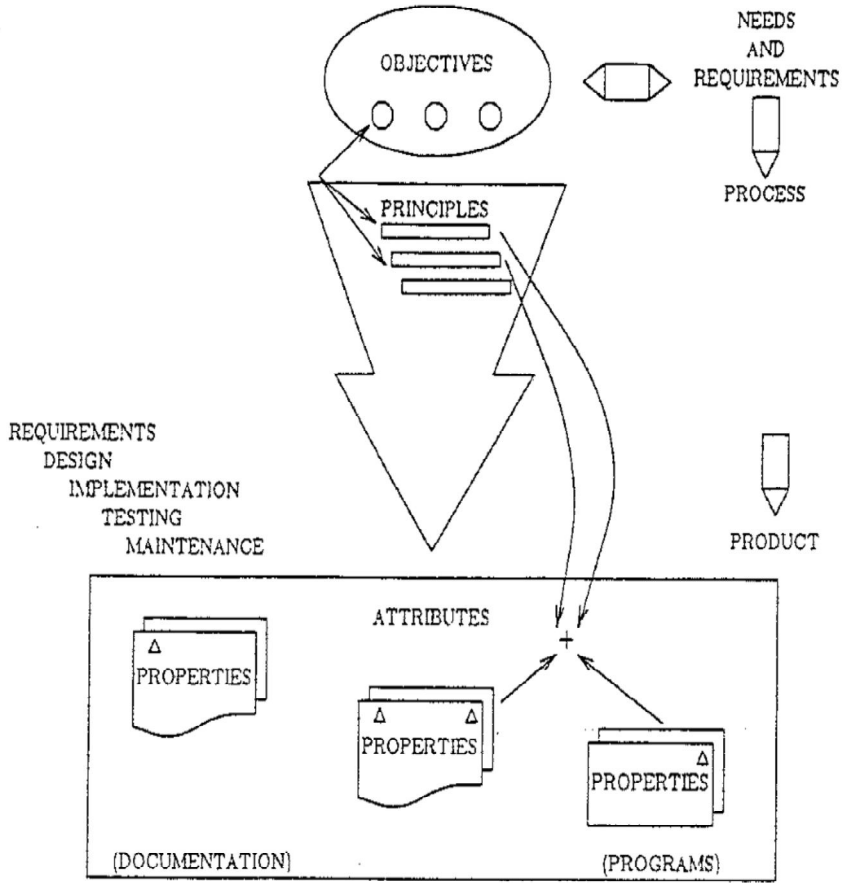
The black box testing allows the testing person to consider each code module as an entire unit with defined inputs and outputs and not considering how the input is modified to output. With the help of this method there is no need to study the internal operations of the code and the requirement of source code is not necessary. Typical example for black box testing is boundary value analysis where we enter then input values and we get the output values which are valid or invalid depending on the boundary values. Generally black box testing is performed after complete integration of system or when the entire module for code is done.

One of the best methods for performing evaluation of COTS based products the black box testing is scenario based testing. The scenarios are based on the requirements and procedures which the system has to perform. Procedures for the test are organised and each component is tested compared to criteria.

Volume 4 Issue XI, November 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

V. EVALUATION OF SOFTWARE DEVELOPMENT METHODOLOGIES

The software development process contains objectives, principles and attributes.



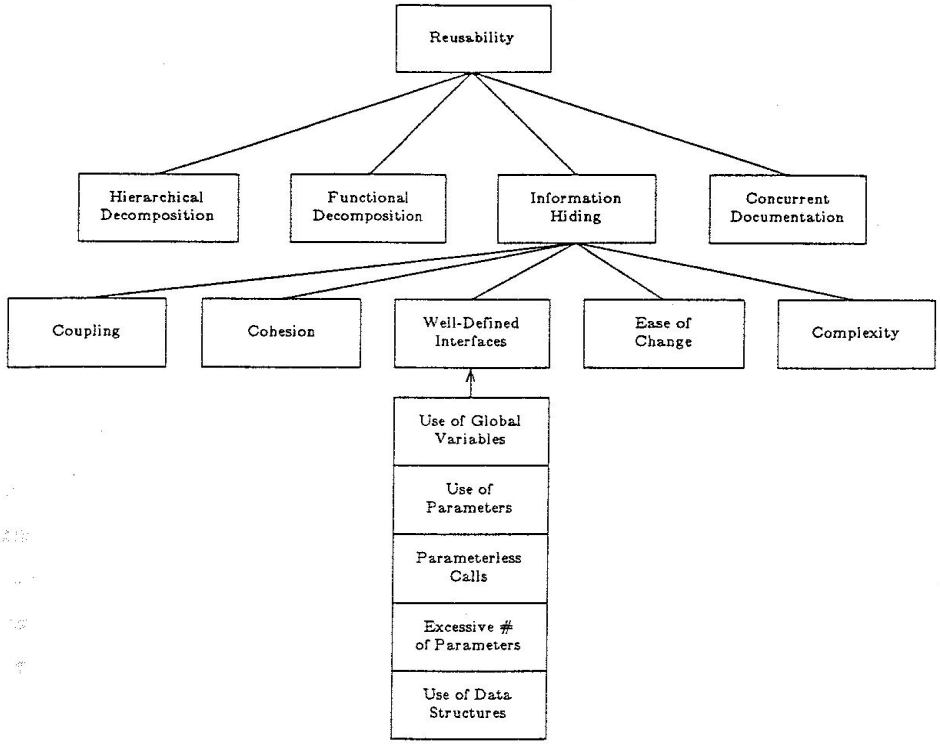
Generally objectives are goals or desirables related to project which can only achieved when the project is completed. The principles related to software engineering tells us how a particular software should be developed. For this to happen one should always keep in mind about the objectives for the software development. At the start of the project these principles and rules may be unclear, but recently many principles have been developed in software development literature. Attributes are immaterial features. Attributes can be displayed by each component of code, but their immaterial nature makes them even difficult to feel its presence.

A. *Structure for evaluation*

1) *Maintainability:* The ability to rectify the contingencies with ease.
2) *Correctness:* The ability to check whether the requirements are as per the specifications.
3) *Reusability:* The ability to reuse a developed software in a particular application.
4) *Testability:* The ability to evaluate all the specifications which the software should exhibit.
5) *Reliability:* The ability to produce performance without any discrepancies over a period of time.
6) *Portability:* The ability to transfer a particular software product from one system to another.
7) *Adaptability:* The ability of the software to withstand any changes over a period of time.

The evaluation process is done by considering a particular objective, Reusability. It is split in four principles namely hierarchical decomposition, functional decomposition, information hiding and concurrent documentation. Again information hiding is split into five attributes namely coupling, cohesion, well defined interfaces, ease of change, complexity. With these divisions for a particular objective we can easily decide what type of behaviour a particular software can exhibit. These concepts are well explained in the diagram that follows.

Volume 4 Issue XI, November 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)



VI. LATEST ADVANCEMENTS IN SOFTWARE TESTING

With the progress in research and development on various types of software testing emphasis is more on component system testing where flexibility and repeatability of the system can be improved.

Latest strategies for software testing are as follows:

GUI automation test:

The testing framework is automated which adopts object-oriented technology for GUI. The test cases are generated automatically by adopting ant colony algorithm. With the help of this algorithm and bit coding a model for input is developed and ant paths for the algorithm is created. Due to this a number of ant paths are improved and the degree of immobility of immobility.

Component software testing:

The software components testability is based on the following factors:

A. Controllability

The extent to which the component under test (CUT) can be controlled.

B. Observability

The extent to which the test results can be observed.

C. Isolateability

The extent to which isolation can be performed on the component under test (CUT).

D. Understandability

The extent to which the component under test (CUT) can be understood.

E. Automatability

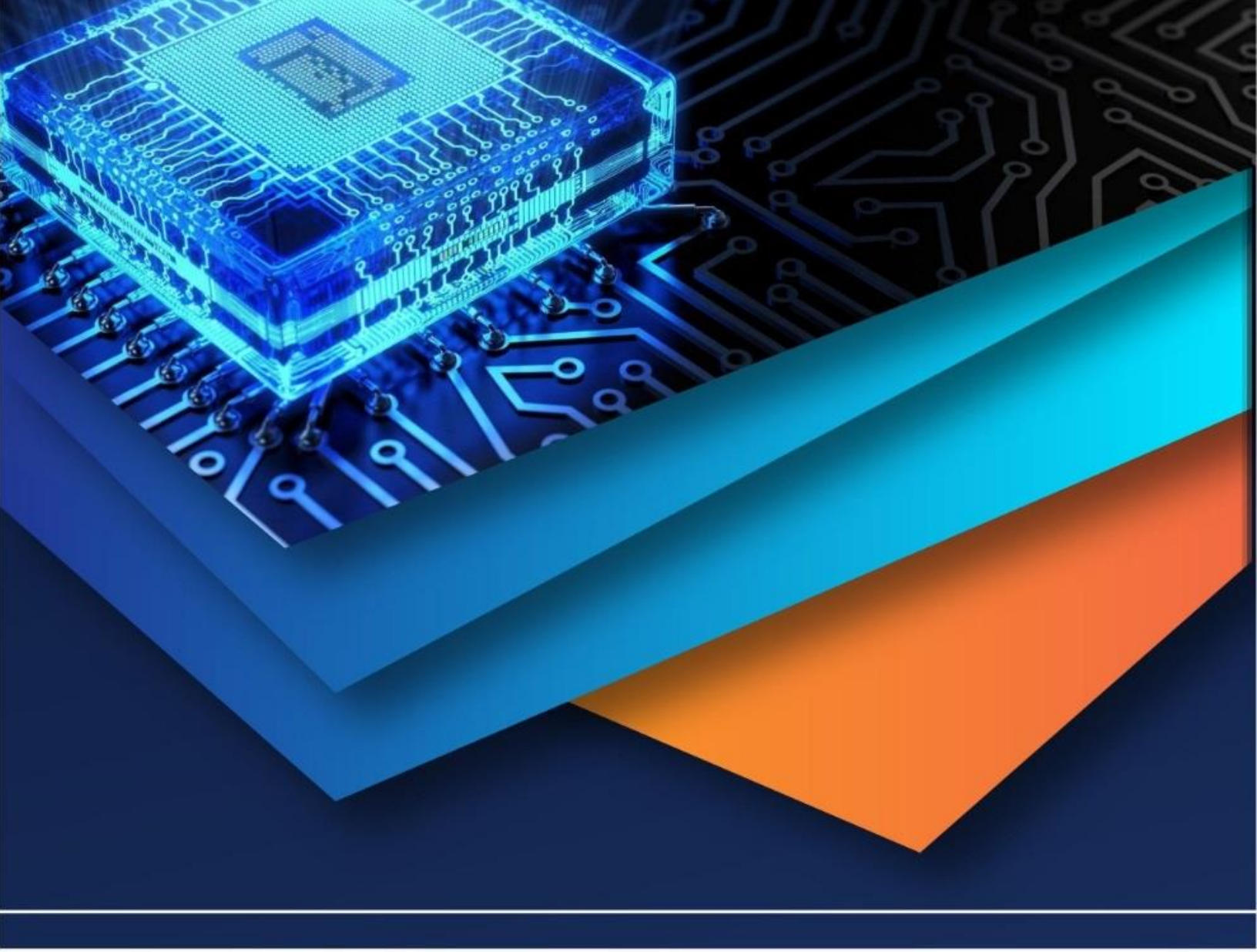
The extent to which the component under test (CUT) is subjected to automation.

VII. CONCLUSION

Software evaluation involves proper evaluators and appropriate selection criteria. The emphasis should be on to how much extent are the variations in scoring by the respective evaluators and selection criteria should be based on the requirements. The risk management should be considered to alleviate the failures of software projects by using regression techniques. By qualitative weight and sum method we can obviously describe the evaluation, understand the logic behind the evaluation and knowing the possible limitations. It is feasible to design an evaluation criteria which is theoretically based for a particular purpose and goal. The methods discussed above provides a platform for choosing relevant COTS software products which can be used in large scale systems. We have discussed about the black box and white box testing. In procedural approach to software development we have discussed about the objectives, principles and attributes and the relation among them. This approach could be a promising effort for the evaluation of software project methodologies.

REFERENCES

[1] Bandor, M. S. (2006). Quantitative methods for software selection and evaluation (No. CMU/SEI-2006-TN-026). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
[2] Elzamly, A., & Hussin, B. (2015). Modelling and Evaluating Software Project Risks with Quantitative Analysis Techniques in Planning Software Development. CIT. Journal of Computing and Information Technology, 23(2), 123-139.
[3] Baumgartner, P., & Payr, S. (1997). Methods and practice of software evaluation. The case of the European Academic Software Award. In Proceedings of ED-MEDIA 97–World Conference on Educational Multimedia and Hypermedia.
[4] Dean, J. C., Se, B. S., Math, M., & Vigder, M. R. (2000). COTS software evaluation techniques. NATIONAL RESEARCH COUNCIL OF CANADA OTTAWA (ONTARIO) SOFTWARE ENGINEERING GROUP.
[5] Arthur, J. D., Nance, R. E., & Henry, S. M. (1986). A procedural approach to evaluating software development methodologies: The foundation.
[6] Chauhan, R. K., & Singh, I. (2014). Latest Research and Development on Software Testing Techniques and Tools. International Journal of Current Engineering and Technology, 4(4).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY