Multi-attribute Comparison of Automated Functional and Regression Testing Tools using Fuzzy AHP

Praveen Ranjan Srivastava¹, Mahesh Prasad Ray²

¹PhD student, ²ME student
Computer Science & Information System Group, BITS PILANI – 333031 (INDIA)
{praveenrsrivastava, maheshray123}@gmail.com

Source: Proceedings of the 4th Indian International Conference on Artificial Intelligence, IICAI 2009, Tumkur, Karnataka, India, December 16-18, 2009; 01/2009 (INDEXING IN DBLP and SCOUPS)

Abstract. Software Testing is a process used to help in improving quality and efficiency of a software product. In order to make the testing process less time consuming and efficient Automated Testing Tools are being used. The correct choice of an automated testing tool is a critical success factor for the developed product to reach and maintain market leadership. In the current paper a model for selecting an automated functional and regression testing tool using the Fuzzy Analytical Hierarchy Process (FAHP) is presented. The FAHP is used to compare the Testing Tools. The means of the triangular Fuzzy numbers produced by the experts from different CMM level 5 organizations were successfully used in the pair-wise comparison matrix.

Keywords: Software Testing; Automated Testing Tool; Functional and Regression Testing Tool; Fuzzy Analytic Hierarchy Process (FAHP); Multi-criteria Decision Analysis (MCDA).

1 Introduction

Software Testing is a process used to help identify the correctness, completeness and quality of developed computer software [1]. It is a series of processes conducted to provide stakeholders with information about the quality of the product or service under test [2], with respect to the particular environment in which it is intended to operate. Software Testing is one of the most time, effort and resource consuming stage of the software development life cycle as it roughly consumes half of the total cost spent on developing the particular project. To optimize the cost and time of Testing Testers go for automated testing instead of manual testing process. The evolution of automated testing tools for software and applications has been accelerating at a rapid pace and the number of available products on the market has grown significantly. These product choices are accompanied by a dizzying set of product features leading to software testing tools that are available at many different
levels of sophistication and prices. Hence the most appropriate tool should be chosen, which will make the testing process cost effective [3] efficient and less time consuming.

As the available feature set is so rich, and the price range is so wide, it is important for the project managers to choose the most appropriate testing tool for their projects. While it is often assumed that any commercial software testing tool will perform basic functions (and hence, a choice based on price alone is made), prospective buyers need to perform a careful selection analysis to accurately evaluate the feature sets of the many available tools on the market as per their requirements. Given the current state of interest in software testing tools, surprisingly limited work has been done in this area and little attention has been paid to the evaluation of the selection criteria.

2 Background

Although the selection of testing Tool is an important issue in optimizing the cost of testing as well as product development cost but surprisingly limited work has been done in this area and little attention has been paid to the evaluation of the selection criteria. Looking at the present scenario several works has been done in the area of software project management [4] in comparison to software testing. Most of the previous empirical research and investigation in software testing focuses on improvement in performance of testing processes [5] and achievement of quality using automated testing tools [6] along with identification of metrics for measuring the effectiveness of software-testing tools [7] has been done. There is no relevant work found on software testing tool selection on the basis of certain evaluation criteria. The comparison of Testing Tools is done by considering their extent of achieving the attributes in a Fuzzy Linguistic method. So, this work will establish a framework for comparing the automated functional and regression testing tools which can help not only testers and project managers within an organization to select the tools as per their requirements but also to the testing tool vendors in signifying the short-comings in their products and improving product quality in the succeeding versions. Hence the Fuzzy Linguistic method is developed in determining the best of all available tools.

The present paper is mainly intended to underline the necessity of dealing with uncertainty to manage the risk of decision correctly and to propose a solution by fuzzy mathematics. The paper illustrates some key aspects of Software Testing Fuzzy Evaluation Method (STFEM), an integrated evaluation methodology. It is mainly focused around an integrated definition of the evaluation process, the application of a new class of fuzzy aggregators of Ordered Fuzzy Number Weighted Averaging. The paper shows some results of the application of this method to a concrete study-case.

3 The Fuzzy Analytic Hierarchy Process (FAHP)

The Analytic Hierarchy Process (AHP) was developed in the early 1980’s by Thomas
Saaty [8] to solve prioritization problems. The Analytic Hierarchy Process (AHP) [9] can be served as a powerful tool in calculating weights and pursuing MCDA procedure. The AHP is able to select the best alternative in a Multi-Hierarchical System. Satty claims that the AHP serves as a framework for people to structure their own problems and provide judgments based on knowledge and experience. Some of its applications of AHP include Transport Planning in the Sudan [10], Choosing a Modern Computer System [11] and Political Candidacy [12]. It is difficult to map qualitative preferences to point estimates a degree of uncertainty will be associated with some or all pair-wise comparison, which is called fuzzy AHP problem.


In order to deal with the uncertainty and vagueness from subjective perceptions and experience of human in the decision process a methodology based on the scale given in Table 1 by Chang [16] extent fuzzy AHP modeling to access the tangible and intangible balance is proposed.

<table>
<thead>
<tr>
<th>Linguistic form of comparison</th>
<th>Priority of one over other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute</td>
<td>(7/2, 4, 9/2)</td>
</tr>
<tr>
<td>Very Strong</td>
<td>(5/2, 3, 7/2)</td>
</tr>
<tr>
<td>Fairly Strong</td>
<td>(3/2, 2, 5/2)</td>
</tr>
<tr>
<td>Weak</td>
<td>(2/3, 1, 3/2)</td>
</tr>
<tr>
<td>Equal</td>
<td>(1, 1, 1)</td>
</tr>
</tbody>
</table>

Let, X = \{x_1, x_2, x_3, \ldots, x_n\} be an object set, and U = \{u_1, u_2, u_3, \ldots, u_n\} be a goal set. According to the method of Chang’s (1992) extent analysis, each object is taken and extent analysis for each goal \(g_i\) is performed respectively. Therefore, m extent analysis values for each object can be obtained, with the following signs:

\[ M_{g_i}^1, M_{g_i}^2, \ldots, M_{g_i}^m, i = 1, 2, \ldots, n \]

Where all the \(M_{g_i}^j\) (j = 1, 2, \ldots, m) are Triangular Fuzzy Numbers.

The steps of Chang’s extent analysis [14] can be given as in the following:
Step 1:

The value of fuzzy synthetic extent with respect to the $i_{th}$ object is defined as,

$$S_i = \sum_{j=1}^{m} M_{gi}j \odot \left[ \sum_{j=1}^{n} \sum_{i=1}^{m} M_{gi}j \right]^{-1} \sum_{j=1}^{m} M_{gi}j$$

To obtain $\sum_{j=1}^{m} M_{gi}j$, perform the fuzzy addition operation of $m$ extent analysis values for a particular matrix such as,

$$\sum_{j=1}^{m} M_{gi}j = \left( \sum_{j=1}^{m} l_j, \sum_{j=1}^{m} m_j, \sum_{j=1}^{m} u_j \right)$$

And to obtain $\sum_{j=1}^{n} \sum_{i=1}^{m} M_{gi}j$, perform the fuzzy addition operation of $M_{gi}j$ $(j=1, 2,..m)$ values such that,

$$\sum_{i=1}^{n} \sum_{j=1}^{m} M_{gi}j = \left( \sum_{j=1}^{n} l_j, \sum_{j=1}^{n} m_j, \sum_{j=1}^{n} u_j \right)$$

And then compute the inverse of the vector in above equation such that,

$$\left[ \sum_{i=1}^{n} \sum_{j=1}^{m} M_{gi}j \right]^{-1} = \left( \frac{1}{\sum_{j=1}^{n} u_j}, \frac{1}{\sum_{j=1}^{n} m_j}, \frac{1}{\sum_{j=1}^{n} l_j} \right)$$

Step 2:

The degree of possibility of $M_2 = (l_2, m_2, u_2) \geq M_1 = (l_1, m_1, u_1)$ is defined as,

$$V(M_2 \geq M_1) = \left( \forall x \geq x_{M_1} \min \left( \mu_{M_2}(x), \mu_{M_1}(x) \right) \right)$$

and can be equivalently expressed as follows:

$$V(M_2 \geq M_1) = \text{hgt}(M_1 \cap M_2) = \mu_{M_2}(a)$$

$$= \left\{ \begin{array}{ll}
1, & \text{if } m_2 \geq m_1, \\
0, & \text{if } l_1 \geq u_2, \\
\frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, & \text{otherwise},
\end{array} \right.$$
Where \( d \) is the ordinate of the highest intersection point \( D \) between \( \mu_{M_1} \) and \( \mu_{M_2} \).

To compare \( M_1 \) and \( M_2 \), we need both the values of

\[
V(M_1 \geq M_2) \quad \text{and} \quad V(M_2 \geq M_1).
\]

Step 3:

From the figure I, the intersection of \( M_1 \) and \( M_2 \) can be determined, which helps to calculate the degree possibility for a convex fuzzy number to be greater than \( k \) convex fuzzy numbers \( M_i \) \((1, 2…k)\) can be defined by,

\[
V(M \geq M_i, M_2…M_k) = V(M \geq M_i) \quad \text{and} \quad V(M \geq M_2) \quad \text{and} \quad V(M \geq M_k).
\]

\[
= \min V(M \geq M_i), i = 1, 2, 3,…k
\]

Assume that \( d(A_i) = \min V(S_i \geq S_k) \)

For \( k = 1, 2…n; k \neq i \). Then the weight factor is given by,

\[
W' = (d(A_1), d'(A_2), d'(A_n))
\]

Where \( A_i \) \((i = 1, 2,…n)\) are \( n \) elements.

Step 4:

Via normalization, the normalized weight vectors are,

\[
W = (d(A_1), d(A_2), d(A_n))
\]

Where \( W \) is a non-fuzzy number.

![Fig. 1. The Intersection between \( M_1 \) and \( M_2 \)](image)

4 Case Study

In this paper an application of the FAHP methodology, applied to different types of functional and regression testing tools are available [16] in the market among those six popular automated functional and regression testing tools are taken. These tools are Win Runner, QA Run, Silk Test, Visual Test, Robot and QA Wizard. To make comparisons between them the above attributes are considered and compared with each other to determine the relative importance of each factor with regards to the overall goal.

![Fig 2. Multi-attribute Selection](image)
5 Evaluation Criteria

A rich set of testing tools are used by the testers in automated functional and regression testing to make the project free of bug [17]. It is true that most commercial functional and regression testing tools will perform the basic functions of debugging [18]. Hence, a choice of testing tools should not be done based on price alone. Prospective buyers need to perform a careful selection analysis to accurately evaluate the feature sets of the many available tools on the market, according to their requirements. In the current paper ten qualitative and quantitative criteria are taken on consulting experts [19] from testing units of different CMM level 5 software organizations to compare and evaluate software testing tools as shown in Table 2. The criteria chosen are by no means the only ones possible. This set of criteria is selected, however, because it was identified as suitable for making software testing tools comparisons as these are the most influencing ones.

Table 2. Comparison of Testing Tools

<table>
<thead>
<tr>
<th>Pairwise_Compare</th>
<th>Record/Playback</th>
<th>Environment Support</th>
<th>Cost</th>
<th>Ease Of Use</th>
<th>Database Tests</th>
<th>Object Mapping</th>
<th>Test/Error Recovery</th>
<th>Support from Respective Vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>WiseRunner</td>
<td>1.1</td>
<td>23.1/23</td>
<td>52.3/12</td>
<td>52.5/2</td>
<td>70.4/2</td>
<td>23.1/23</td>
<td>70.4/2</td>
<td>23.1/23</td>
</tr>
<tr>
<td>QA Run</td>
<td>23.1/23</td>
<td>23.1/23</td>
<td>52.3/12</td>
<td>52.5/2</td>
<td>70.4/2</td>
<td>23.1/23</td>
<td>70.4/2</td>
<td>23.1/23</td>
</tr>
<tr>
<td>Silk Test</td>
<td>27.1/27</td>
<td>25.1/23</td>
<td>1.1</td>
<td>52.3/12</td>
<td>70.4/2</td>
<td>23.1/23</td>
<td>70.4/2</td>
<td>23.1/23</td>
</tr>
<tr>
<td>Visual Test</td>
<td>25.1/23</td>
<td>27.1/27</td>
<td>1.1</td>
<td>52.3/12</td>
<td>70.4/2</td>
<td>23.1/23</td>
<td>70.4/2</td>
<td>23.1/23</td>
</tr>
<tr>
<td>Robot</td>
<td>29.1/29</td>
<td>25.1/23</td>
<td>1.1</td>
<td>52.3/12</td>
<td>70.4/2</td>
<td>23.1/23</td>
<td>70.4/2</td>
<td>23.1/23</td>
</tr>
<tr>
<td>QA Wizard</td>
<td>23.1/23</td>
<td>25.1/23</td>
<td>1.1</td>
<td>52.3/12</td>
<td>70.4/2</td>
<td>23.1/23</td>
<td>70.4/2</td>
<td>23.1/23</td>
</tr>
</tbody>
</table>

The evaluated attributes are given in the Figure 2 as, (1) Environment Support, (2) Cost, (3) Ease of Use, (4) Database Tests, (5) Object Mapping, (6) Extensible Language, (7) Integration, (8) Test/Error Recovery, (9) Record and Playback, (10) Support from Respective Vendors. In this paper a fuzzy comparison is made in order to find the best alternatives from the available rich set of testing tools. As per the user or tester’s requirement he has to give the preference order of each attributes over
other. The users have to give their preference order in linguistic form. From the given input the Table 3 can be obtained by using Chang’s Fuzzy Linguistic Scale as given in Table 1.

Table 3. Pair-wise Comparison of Attributes

<table>
<thead>
<tr>
<th>Pairwise Comp.</th>
<th>Record &amp; Playback</th>
<th>Environment Support</th>
<th>Cost</th>
<th>Ease Of Use</th>
<th>Database Tests</th>
<th>Object Mapping</th>
<th>Extensible Language</th>
<th>Integration</th>
<th>Test/Err recovery</th>
<th>Vendor Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record &amp; Playback</td>
<td>1.1</td>
<td>23.1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
<td>7/2, 4, 9/2</td>
<td>2/3, 1, 3/2</td>
<td>7/2, 4, 9/2</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
</tr>
<tr>
<td>Environment Support</td>
<td>2/3, 1, 3/2</td>
<td>1.1</td>
<td>3/2, 2, 5/2</td>
<td>5/2, 3, 7/2</td>
<td>7/2, 4, 9/2</td>
<td>3/2, 2, 5/2</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
</tr>
<tr>
<td>Cost</td>
<td>2/7, 1, 3/2</td>
<td>2/5, 1, 2/3</td>
<td>1, 1</td>
<td>1/1, 1</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>3/2, 2, 5/2</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
</tr>
<tr>
<td>Ease Of Use</td>
<td>2/5, 1, 2/3</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>1, 1</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>3/2, 2, 5/2</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
</tr>
<tr>
<td>Database Tests</td>
<td>2/9, 1, 4/2</td>
<td>2/9, 1, 4/2</td>
<td>2/9, 1, 4/2</td>
<td>1/1</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>3/2, 2, 5/2</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
</tr>
<tr>
<td>Object Mapping</td>
<td>2/3, 1, 3/2</td>
<td>2/5, 1, 2/3</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>1, 1</td>
<td>3/2, 2, 5/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
</tr>
<tr>
<td>Ext. Language</td>
<td>2/9, 1, 4/2</td>
<td>2/9, 1, 4/2</td>
<td>2/9, 1, 4/2</td>
<td>2/9, 1, 4/2</td>
<td>1/1</td>
<td>3/2, 2, 5/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
<td>5/2, 3, 7/2</td>
<td>2/3, 1, 3/2</td>
</tr>
<tr>
<td>Integration</td>
<td>2/3, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>1, 1</td>
<td>3/2, 2, 5/2</td>
</tr>
<tr>
<td>Test/Err rec.</td>
<td>2/3, 1, 3/2</td>
<td>2/3, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>1, 1</td>
</tr>
<tr>
<td>Vendor Support</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
<td>2/7, 1, 3/2</td>
</tr>
</tbody>
</table>

Applying the algorithm step wise the best alternative can be obtained.

Step 1:

\[ S (\text{Record and Playback}) = (16.51, 21, 25.5) \overset{147.07}{\underset{138.85}{\rightsquigarrow}} \overset{116.78}{\underset{1}{\left( \frac{1}{147.07}, \frac{1}{138.85}, \frac{1}{116.78} \right)}}, \]

To get the value of \( \sum_{j=1}^{m} l_j \), first element of all the pairs of Record and Playback from Table II are to be added up as, \( 1+2/3+5/2+3/2+7/2+2/3+7/2+2/3+2/3+5/2 = 16.51 \).

And to calculate \( \sum_{j=1}^{n} u_j \), all the first elements of the pairs available from the table II are to be added. Similarly other values can also be calculated to get the relative importance of each factor over other.

\[ S (\text{Environment Support}) = (17.01, 17.50, 25.5) \overset{147.07}{\underset{138.85}{\rightsquigarrow}} \overset{116.78}{\underset{1}{\left( \frac{1}{147.07}, \frac{1}{138.85}, \frac{1}{116.78} \right)}}, \]

\[ S (\text{Cost}) = (14.52, 17.84, 21.57) \overset{147.07}{\underset{138.85}{\rightsquigarrow}} \overset{116.78}{\underset{1}{\left( \frac{1}{147.07}, \frac{1}{138.85}, \frac{1}{116.78} \right)}} \]
\[
S (\text{Ease of Use}) = (15.13, 18.18, 21.47)
\]
\[
S (\text{Database Test}) = (0.103, 0.131, 0.184)
\]
\[
S (\text{Object Mapping}) = (11.86, 14.75, 17.84)
\]
\[
S (\text{Extensible Language Support}) = (0.081, 0.106, 0.153)
\]
\[
S (\text{Integration}) = (13.03, 15.18, 18.64)
\]
\[
S (\text{Test/Error Recovery}) = (0.088, 0.109, 0.160)
\]
\[
S (\text{Vendor Support}) = (4.05, 4.86, 6)
\]

Step 2:

After the Fuzzy extent calculation, the degree of possibilities of each attribute is to be calculated as shown below.

\[
V (\text{Record and Playback} \geq \text{Environment Support}) = 1
\]
\[
V (\text{Record and Playback} \geq \text{Cost}) = 1
\]
\[
V (\text{Record and Playback} \geq \text{Ease of Use}) = 1
\]
\[
V (\text{Record and Playback} \geq \text{Database}) = 1
\]
\[
V (\text{Record and Playback} \geq \text{Object Mapping}) = 1
\]
\[
V (\text{Record and Playback} \geq \text{Extensible Language Support}) = 1
\]
\[
V (\text{Record and Playback} \geq \text{Integration}) = 1
\]
\[
V (\text{Record and Playback} \geq \text{Error Recovery}) = 1
\]
\[
V (\text{Record and Playback} \geq \text{Vendor Support}) = 1
\]
\[
V (\text{Environment Support} \geq \text{Record and Playback}) = 0.809
\]
\[
V (\text{Environment Support} \geq \text{Cost}) = \frac{0.098 - 0.218}{0.126 - 0.218} = 0.984
\]
\[
V \ (\text{Environment Support} \geq \text{Ease of Use}) = \frac{(0.126 - 0.218) - (0.131 - 0.103)}{0.958}
\]

\[
V \ (\text{Environment Support} \geq \text{Database}) = 1
\]

\[
V \ (\text{Environment Support} \geq \text{Object Mapping}) = 1
\]

\[
V \ (\text{Environment Support} \geq \text{Extensible Language Support}) = 1
\]

\[
V \ (\text{Environment Support} \geq \text{Integration}) = 1
\]

\[
V \ (\text{Environment Support} \geq \text{Error Recovery}) = 1
\]

\[
V \ (\text{Environment Support} \geq \text{Vendor Support}) = 1
\]

\[
V \ (\text{Cost} \geq \text{Record and Playback}) = 0.758
\]

\[
V \ (\text{Cost} \geq \text{Environment Support}) = 1
\]

\[
V \ (\text{Cost} \geq \text{Ease of Use}) = 0.964
\]

\[
V \ (\text{Cost} \geq \text{Database}) = 1
\]

\[
V \ (\text{Cost} \geq \text{Object Mapping}) = 1
\]

\[
V \ (\text{Cost} \geq \text{Extensible Language Support}) = 1
\]

\[
V \ (\text{Cost} \geq \text{Integration}) = 1
\]

\[
V \ (\text{Cost} \geq \text{Error Recovery}) = 1
\]

\[
V \ (\text{Cost} \geq \text{Vendor Support}) = 1
\]

\[
V \ (\text{Ease of Use} \geq \text{Record and Playback}) = 0.783
\]

\[
V \ (\text{Ease of Use} \geq \text{Environment Support}) = 1
\]

\[
V \ (\text{Ease of Use} \geq \text{Cost}) = 1
\]

\[
V \ (\text{Ease of Use} \geq \text{Database}) = 1
\]

\[
V \ (\text{Ease of Use} \geq \text{Object Mapping}) = 1
\]

\[
V \ (\text{Ease of Use} \geq \text{Extensible Language Support}) = 1
\]

\[
V \ (\text{Ease of Use} \geq \text{Integration}) = 1
\]

\[
V \ (\text{Ease of Use} \geq \text{Error Recovery}) = 1
\]

\[
V \ (\text{Ease of Use} \geq \text{Vendor Support}) = 1
\]

\[
V \ (\text{Database} \geq \text{Record and Playback}) = 0.477
\]

\[
V \ (\text{Database} \geq \text{Environment Support}) = \frac{(0.106 - 0.153) - (0.126 - 0.115)}{0.655} = 0.098 - 0.153
\]

\[
V \ (\text{Database} \geq \text{Cost}) = \frac{(0.106 - 0.153) - (0.128 - 0.098)}{0.714} = 0.103 - 0.153
\]

\[
V \ (\text{Database} \geq \text{Ease of Use}) = \frac{(0.106 - 0.153) - (0.131 - 0.103)}{0.667} = 0.088 - 0.153
\]

\[
V \ (\text{Database} \geq \text{Object Mapping}) = \frac{(0.106 - 0.153) - (0.109 - 0.088)}{0.956}
\]

\[
V \ (\text{Database} \geq \text{Extensible Language Support}) = 1
\]

\[
V \ (\text{Database} \geq \text{Integration}) = 1
\]

\[
V \ (\text{Database} \geq \text{Error Recovery}) = 1
\]

\[
V \ (\text{Database} \geq \text{Vendor Support}) = 1
\]

\[
V \ (\text{Object Mapping} \geq \text{Record and Playback}) = 0.534
\]

\[
V \ (\text{Object Mapping} \geq \text{Environment Support}) = 0.726
\]
\[ V(\text{Object Mapping} \geq \text{Cost}) = 0.765 \]
\[ V(\text{Object Mapping} \geq \text{Ease of Use}) = 0.722 \]
\[ V(\text{Object Mapping} \geq \text{Database}) = 1 \]
\[ V(\text{Object Mapping} \geq \text{Extensible Language Support}) = 1 \]
\[ V(\text{Object Mapping} \geq \text{Integration}) = 1 \]
\[ V(\text{Object Mapping} \geq \text{Error Recovery}) = 1 \]
\[ V(\text{Object Mapping} \geq \text{Vendor Support}) = 1 \]
\[ V(\text{Extensible Language Support} \geq \text{Record and Playback}) = 0.278 \]
\[ V(\text{Extensible Language Support} \geq \text{Environment Support}) = 0.373 \]
\[ V(\text{Extensible Language Support} \geq \text{Cost}) = 0.514 \]
\[ V(\text{Extensible Language Support} \geq \text{Ease of Use}) = 0.456 \]
\[ V(\text{Extensible Language Support} \geq \text{Database}) = 0.815 \]
\[ V(\text{Extensible Language Support} \geq \text{Object Mapping}) = 0.754 \]
\[ V(\text{Extensible Language Support} \geq \text{Integration}) = 1 \]
\[ V(\text{Extensible Language Support} \geq \text{Error Recovery}) = 1 \]
\[ V(\text{Extensible Language Support} \geq \text{Vendor Support}) = 1 \]
\[ V(\text{Integration} \geq \text{Record and Playback}) = 0 \]
\[ V(\text{Integration} \geq \text{Environment Support}) = 0 \]
\[ V(\text{Integration} \geq \text{Cost}) = 0.127 \]
\[ V(\text{Integration} \geq \text{Ease of Use}) = 0.058 \]
\[ V(\text{Integration} \geq \text{Database}) = 0.394 \]
\[ V(\text{Integration} \geq \text{Object Mapping}) = 0.306 \]
\[ V(\text{Integration} \geq \text{Extensible Language Support}) = 0.534 \]
\[ V(\text{Integration} \geq \text{Error Recovery}) = 1 \]
\[ V(\text{Integration} \geq \text{Support}) = 1 \]
\[ V(\text{Error Recovery} \geq \text{Record and Playback}) = 0 \]
\[ V(\text{Error Recovery} \geq \text{Environment Support}) = 0 \]
\[ V(\text{Error Recovery} \geq \text{Cost}) = 0 \]
\[ V(\text{Error Recovery} \geq \text{Ease of Use}) = 0 \]
\[ V(\text{Error Recovery} \geq \text{Database}) = 0 \]
\[ V(\text{Error Recovery} \geq \text{Object Mapping}) = 0 \]
\[ V(\text{Error Recovery} \geq \text{Extensible Language Support}) = 0.087 \]
\[ V(\text{Error Recovery} \geq \text{Integration}) = 0.034 \]
\[ V(\text{Error Recovery} \geq \text{Support}) = 1 \]
\[ V(\text{Vendor Support} \geq \text{Record and Playback}) = 0 \]
\[ V(\text{Vendor Support} \geq \text{Environment Support}) = 0 \]
\[ V(\text{Vendor Support} \geq \text{Cost}) = 0 \]
V (Vendor Support ≥ Ease of Use) = 0
V (Vendor Support ≥ Database) = 0
V (Vendor Support ≥ Object Mapping) = 0
V (Vendor Support ≥ Extensible Language Support) = 0
V (Vendor Support ≥ Integration) = 0
V (Vendor Support ≥ Error Recovery) = 0.414

Step 3:
After calculating the degree of possibilities, the individual values to be normalized. Sum of all values corresponding to Record and Playback is $1+1+1+1+1+1+1+1+1+1+1+1 = 9$. Similarly the sums of respective attributes are to be calculated. And the sum of all values is found to be 61.62. So, priority vector correspond to Record and Playback is $9/61.62 = 0.146$ and similarly other values can be calculated and the priority vectors of respective attributes are placed in Table III. The non-fuzzy number $W_{\text{Attribute}} = (0.146, 0.142, 0.141, 0.143, 0.121, 0.126, 0.100, 0.055, 0.018, 0.007)^T$ is the weight vector of attributes.

Table 4. Priority Vector

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Priority Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record &amp; Playback</td>
<td>0.146</td>
</tr>
<tr>
<td>Environment Support</td>
<td>0.142</td>
</tr>
<tr>
<td>Cost</td>
<td>0.141</td>
</tr>
<tr>
<td>Ease Of Use</td>
<td>0.143</td>
</tr>
<tr>
<td>Database Tests</td>
<td>0.121</td>
</tr>
<tr>
<td>Object Mapping</td>
<td>0.126</td>
</tr>
<tr>
<td>Extensible Language</td>
<td>0.100</td>
</tr>
<tr>
<td>Integration</td>
<td>0.055</td>
</tr>
<tr>
<td>Test/Error recovery</td>
<td>0.018</td>
</tr>
<tr>
<td>Support</td>
<td>0.007</td>
</tr>
</tbody>
</table>

After calculating the weight vector of attributes, the weight vector of testing tools are to be calculated. For that pair-wise comparison of testing tools on the basis of individual attributes are to be done from the above comparison table 2. Table 5 shows pair-wise comparison of testing tools with respect to the attribute Record and Playback. Applying the same procedure the non-fuzzy numbers $W_i = (0.256, 0.165, 0.142, 0.141, 0.158, 0.127)^T$ is the weight Vector of tools with respect to Record and Playback is calculated.

Table 5. Pair-wise Comparison of Testing Tools

<table>
<thead>
<tr>
<th>W.R.T</th>
<th>Win Runner</th>
<th>QA Run</th>
<th>Silk Test</th>
<th>Visual Test</th>
<th>Robot</th>
<th>QA Wizard</th>
</tr>
</thead>
</table>
Similarly, The non-fuzzy numbers \( W_k \) (k=2….10) the weight Vector of tools with respect to individual attributes are to be calculated. The non-fuzzy number \( W_2 = (0.156, 0.135, 0.145, 0.144, 0.128, 0.167) \) is the weight Vector of tools with respect to Environment Support. The non-fuzzy number \( W_3 = (0.216, 0.175, 0.147, 0.111, 0.151, 0.125) \) is the weight Vector of tools with respect to Cost. The non-fuzzy number \( W_4 = (0.246, 0.265, 0.143, 0.142, 0.151, 0.126) \) is the weight Vector of tools with respect to Ease of Use. The non-fuzzy number \( W_5 = (0.153, 0.135, 0.151, 0.181, 0.138, 0.217) \) is the weight Vector of tools with respect to database Support. The non-fuzzy number \( W_6 = (0.151, 0.132, 0.173, 0.121, 0.149, 0.221) \) is the weight Vector of tools with respect to Object Mapping. The non-fuzzy number \( W_7 = (0.196, 0.175, 0.151, 0.132, 0.191, 0.171) \) is the weight Vector of tools with respect to Extensible Language Support. The non-fuzzy number \( W_8 = (0.154, 0.175, 0.243, 0.253, 0.231, 0.197) \) is the weight Vector of tools with respect to integration. The non-fuzzy number \( W_9 = (0.176, 0.135, 0.165, 0.182, 0.238, 0.215) \) is the weight Vector of tools with respect to Error Recovery. The non-fuzzy number \( W_{10} = (0.218, 0.131, 0.121, 0.152, 0.234, 0.213) \) is the weight Vector of tools with respect to Support.

Hence, finally after obtaining the weight vectors the overall priority of Testing Tools is calculated by,

\[
\text{Result} = [W_1 W_2 W_3 W_4 W_5 W_6 W_7 W_8 W_9 W_{10}] \times W_{\text{Attribute}},
\]

as shown in Table 6. Hence according to the above references given by the user Win Runner is the best alternative out of the all.

<table>
<thead>
<tr>
<th>Automated Testing Tools</th>
<th>Priority Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win Runner</td>
<td>0.195</td>
</tr>
<tr>
<td>QA Run</td>
<td>0.169</td>
</tr>
<tr>
<td>Silk Test</td>
<td>0.155</td>
</tr>
</tbody>
</table>
5 Summary and Conclusion

In this Paper, automated Functional and Regression Testing Tools were compared using Fuzzy AHP. Humans are often uncertain in assigning the evaluation scores in crisp AHP. Fuzzy AHP can capture this difficulty. This work can enhance the process of testing by helping the testers as well as the project managers within software organizations as a result the expenditure in the most costly process of the software life cycle method can be optimized. The goal of this work was not to create a definitive set of features or results, but rather a representative one to illustrate the process. Accordingly, the FAHP helps the software project manager and software tester to identify the principal competitors of the software testing tool and to assess the performance of the tool relative to its competitors as a result the software testing tool vendors or providers can signify the short-comings in their products and can improve their product quality in succeeding versions.

References