
Kavyashree N1,*, Supriya M C2, Lokesh M R3

1 SSAHE Tumkur, Dept of MCA, Dr.AIT, Bangalore
2 Dept of MCA, SSIT, Tumkur
3 Dept of CSE, MITM, Mandya
*Corresponding author. Email: kavyashree1283@gmail.com

ABSTRACT
In small and medium scale software industry, software security success factor is trivial process. Six-sigma and fuzzy logic methodology address software security problems effectively in the state of work. The research work chosen Fuzzy Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and Analytical Hierarchical Process (AHP) method to identify critical success factor estimation. The experimental case study, shows the Euclidean distance amongst the nine components of IOS mobile application. The results discuss on the goal alternate and the best/worst alternate Ranking of the factors to verify mobile IOS application software security. For attaining supportable software security, it is authoritative to safeguard that best practices are conscripted in considering the security valuation at the actual commencement of the software development life cycle.

Keywords: Analytical Hierarchical Process (AHP), critical success factor estimation, IOS mobile application, small and medium scale industry, software security and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS).

1. INTRODUCTION
The strong inexpensive business situation grades in small-medium sized industry pointing for plans to initiative rate decrease and rise the security for the product [1]. Though the goals of the establishments struggle for, small- medium sized industry necessity is to achieve these constant development aims with restricted incomes. The addition of security tools with the Six Sigma methodology is basically significant in this software security development. This article defines the request of the Six Sigma methodology in a Bangalore region small- and medium-sized industry [2].

The software security for small-medium scale industry grants a highway for learning the manufacturing sequence using the DMAIC process (define, measure, analyse, improve, and control). By means of marketplace investigation learning, the team qualifiedly states the problematic, measures the procedure, analyses the supportive data, and implements a sequence of results in improved client contentment. In accumulation, plans are applied to govern the software security developments.

Figure 1 Importance of six sigma in software security for small and medium scale industry

Amongst these fluctuations in the rising struggle and fast planned placing amongst industries. Figure 1 shows the inclination has strengthened in the current times, consequential in industry needing to improve the software efficiency and software security parameters for the development of small and medium scale industry.
Small & medium-scale industry now a day's are
underneath occurrence from outside intimidations and
facts threats. Figure 2. Shows the absence of complexity rund most small and medium scale industry security
posture, the view of enduring natural by occurrences is
miserable [3]. Now small and medium scale industry can
rapidly approve a new skill to improve the new abilities,
new proficiency and decrease the price. Though,
individual new submission makes a necessary to secure
the user data, and the location that the result assimilates
[4]. The finest protecting plan wants to be legalized with
in the period of time. ‘Sense and respond’ must be used
to safeguard preventive actions are employed –seeing
and responding to irregular or doubtful action. The
best protecting plan wants to be legalized with in the
period of time. ‘Sense and respond’ must be used to
safeguard preventive actions are employed – seeing
and responding to irregular or doubtful action. All small
and medium scale industry ght for absence of
time and possessions. They are distant improved off
consecutively and nursing results that
proposition automatic controls in adding to risk
documentation and real time reply.

### Figure 2 Draw back in small and medium scale industry

This unremitting software security development has
been efficiently used by several huge industries but today
it has been newly measured as a possibly operative plan
for several small and medium scale industry [7]. Software
security is progressively observed as a significant pointer
of accomplishment for small-scale industry. Software
security development frequently grades in more for the
business charges but can also result in the development
of client facility and fulfilment [8].

In this paper we primarily highlight on variation of the
accurate software security project in small-medium scale
industry, which is constantly the maximum vital
responsibilities in the effective performance of the
software in any corporate sector. In the present problem
around are not having pure margins in opinion of decision
makers whereas in choosing software security projects in
small-medium scale industry. So, it is important to
evaluate the extreme product in terms of choosing right
security software using decision making technique. In
this framework, a case study of IOS Mobile Application

using fuzzy TOPSIS decision making is used to select the
security for the IOS mobile application.

The rest of this paper is arranged as follows: In
section 2 we provide the related works, software
security problems, software security for mobile
application, for software security TOPSIS, section 3 :
Methodology for the software secure system, section 4 :
Usable security of IOS software for mobile application
Mathematical Model using TOPSIS section 5: A case
study of Mobile Application section 6: Numerical
Analysis of IOS software for Mobile Application using
TOPSIS, section 7: Comparison study with AHP and
TOPSIS, section 8: Conclusion and Future work.

### 2. RELATED WORK

Software security is an awareness applied to guard
software in contradiction of nasty outbreak and extra
hacker perils consequently that the software remains to
role its job properly beneath such possible perils. Security
is required to run veracity, confirmation and accessibility.
Submissions, schemes, and networks are continuously
under numerous security doses such as nasty code or
rejection of package. Approximately the contests from
the submission expansion of security fact of opinion
include Viruses and hackers. The session reviews the
main approaches of software security, software security
using TOPSIS method, and software security for mobile
application.

For software security problems, Software Security
Specifications and Design: How Software Engineers and
Practitioners Are Mixing Things up [1]. Understanding
Software Security from Design to Deployment [2].
Learning Software Security in Context: An Evaluation in
Open Source Software Development Environment [3].
Measurements of the Most Significant Software Security
Weaknesses [4]. The effects of required security on
software development effort [5].

For software using Six-Sigma Quality Management
of Additive Manufacturing [6]. Roadmap for integration
of Lean Six Sigma methodology with ISO 9001: 2015
QMS standard [7]. Measurement System Analysis and
System Thinking in Six Sigma: How They Relate and
How to Use Them [8]. Key criticisms of Six Sigma: a
systematic literature review [9]. Descriptive to Predictive
Six Sigma: Machine Learning for Predictive
Maintenance [10].

For using TOPSIS method in software security, AHP
integrated TOPSIS and VIKOR methods with
Pythagorean fuzzy sets to prioritize risks in self-driving
Pythagorean Fuzzy Hyper soft Environment Based on
Correlation Coefficient and Its Application towards the
Selection of Antivirus Mask in COVID-19 Pandemic
Complexity [12]. A spherical fuzzy methodology

For mobile application in the software security Mobile application security: Mobile application security: Role of perceived privacy as the predictor of security perceptions [17]. A context-aware system using mobile applications and beacons for on-premise security environments [18]. A review of information security aspects of the emerging COVID-19 contact tracing mobile phone applications [19]. Automated security testing of Android applications for secure mobile development [20]. Survey regarding the way students perceive security permissions of mobile applications [21].

3. METHODOLOGY FOR SOFTWARE SECURE SYSTEM FOR SMALL-MEDIUM SCALE INDUSTRIES

The planned method of Software Secure System (SSS) contains of dissimilar stages as shown in Figure 3. Show the Stage of SSS.

![Figure 3 Several phases of SSS](image)

By the way we recognize security is an essential necessity for any software system. This thing has been accomplished by implementing SDLC (Security Development Life Cycle) lecturing security subjects at all the stages. In the current-day world risk organization has to show a vibrant role as the manipulative systems are flatter more and more persistent and Security for the system as they hold serious and personal data. Therefore, it has become significant to design a system that is provenly impervious to the risk they met, as well as it has to overcome weaknesses and vulnerabilities in security device and events to make it more mug proof. This can be accomplished by accepting advanced practices for incoming at most accurate security requirements.

Till this time the work finished on security necessities has assisted us to mitigate the pressures but removal of threats has remained a detached prospect. In this paper we have tried to integrate techniques that assist in scheming an extra secure system which will remove the prospect of disastrous letdown of security device. The recent researches are insufficient to mitigate the unsuccessful state that is unbearable in the predominant security environment. In our new approach we have therefore inducted fuzzy logic to overcome this weakness by making intermediate states in between these two states which will be monitored by actors associated with security mechanism. These actors will continuously sense the security level and will be prompt in taking action once they sense that security level has started moving towards partially secure region. This will avert the threat and so also the failed state. The proposed approach SSS consists of different phases as shown in Figure 3. Phase one consists of elicitation of security requirements compromising identification, categorization, prioritization and mapping of threats into security requirements using Hybrid Technique. Development and application of fuzzy inference system (FIS) to generate the security level of the system at any given point of time has been explained in phase two. Third phase includes association of actors to monitor this security level and adopting additional countermeasures if considered necessary.

4. MATHEMATICAL MODEL OF USABLE-SECURITY ON IOS SOFTWARE USING TOPSIS

TOPSIS, known as Technique for Order of Preference by Similarity to Ideal Solution, is a MCDM. The technique is used in the commercial crossway numerous trades; each period we want to kind a systematic result founded on composed facts.

**TOPSIS Algorithm**

1. Initially, create a matrix that contains the M substitutes and N conditions. The generated matrix is referred as assessment matrix and it is expressed in equation (1).

\[
Q = \begin{pmatrix}
Q_{ij}
\end{pmatrix}
\]

\[M \times N\]

2. Equation (2) shows the normalized assessment matrix.

\[
\alpha_{ij} = \frac{q_{ij}}{\sqrt{\sum_{i=1}^{M} q_{ij}^2}}
\]

Here, each value of j for each i is normalized within the range of 0 and 1. Subsequently, the higher value is considered as better metric.
3. Calculate the weighted normalized resultant matrix. The individual standard considers its own weight and these values are added up to 1. The weights are derivative arbitrarily (i.e., not recommended) or it depends on the skilled information (industry standard).

\[
\chi_{i,j} = \omega_{i,j} \cdot \chi_{i,j}
\]

\[
\omega_{j} = \frac{\omega_{j}}{\sum_{j=1}^{N} \omega_{j}}
\]

\[
\sum_{j=1}^{N} \omega_{j} = 1
\]

Next, the weight is assigned to an individual economic metric for normalizing the values, hence those values are added up to 1. The corresponding normalized weight is multiplied with each normalized value obtained from step 2.

4. For each standard, normalize the best and worst using the equation (6) and (7) respectively.

\[
\chi_{i,j}^{b} = \max_{i=1}^{M} \chi_{i,j}
\]

\[
\chi_{i,j}^{w} = \min_{i=1}^{M} \chi_{i,j}
\]

From these values, the maximum and minimum value for each economic metric are required to be identified in this TOPSIS.

5. Euclidean distance is calculated between the target alternate and the best/worst alternate:

\[
d_{i}^{b} = \sqrt{\sum_{j=1}^{N} (\chi_{i,j} - \chi_{i,j}^{b})^2}
\]

\[
d_{i}^{w} = \sqrt{\sum_{j=1}^{N} (\chi_{i,j} - \chi_{i,j}^{w})^2}
\]

This is an intention of the symmetrical detachment amongst the price of individual economic value for a certain firm i and the best/worst worth of such a value amongst all firms.

6. For each alternate compute the comparison to the poorest alternate. The grades are our TOPSIS marks

\[
s_{i} = \frac{d_{i}^{w}}{d_{i}^{w} + d_{i}^{b}}
\]

We calculate a mark for individual firm which depends on remoteness gained in a step before 7. Rank substitutes giving to the TOPSIS marks by descendent order.

The firm with values nearby to the finest will attain the maximum marks and consequently will be at the top of our ranking. We gained a ranked set of substitutes created on detailed conditions.

5. A CASE STUDY OF MOBILE APPLICATION FOR SMALL AND MEDIUM SCALE INDUSTRY

Mobile applications stand working at comparison and through this degree of progress it is essential that mobile app designers not only express to provide unique and additional structures to the clients but also the security aspects of the application. The below figure-4 shows the software security in IOS mobile application.

Mobile application security is unique and the prime apprehensions as the facts exist within the app [17]. It can be a risk if appropriate security controls are not practically applied while scheming an application [18]. Since there is a more usage of apps in nowadays world mobile application liabilities has enhanced more [19]. Hence designers essentially need to take extra careful while building an app for both IOS and android platforms [21]

Figure 4 Software Security in IOS Mobile Application

Here are some of the ways to build a completely secure mobile application for IOS:

5.1. Write a secure code

Scripting code is the greatest susceptible attribute of any mobile submission which can be broken simply through the intruders [22-28]. Henceforth it is important to script an extremely secured code. Rendering to study around 10.7 million devices are actually being inflated by malevolent code. An intruder can inverse plan besides can bring an app code and practice it in a corrupted method, therefore attempt to build a secure firm code not so easy to disruption and follow agile development so that you can cover and inform your code easily time to time [29].

5.2. Encryption of the data

Encryption is the method to change the data spreading in to such a method that it cannot be delivered by anybody else lacking decryption. This is an effective mode to save the data from existence used in a
malevolent way. So even if the data is drained the intruders cannot decrypt it and use the data.

5.3. While using libraries be careful

Frequently the mobile application code wants the mediator in the public library for the code construction. Do not belief at all library for your application structure as utmost of them be situated not protected. Once you have second hand several types of libraries continuously attempt to examine the encryption [30].

5.4. Use authorized API

Permanently recall to practice authorized API in the application code. It continuously springs hacker’s honor to practice your info for illustration authorization evidence stores can be cast-off via the intruders towards increase authentication happening on the organization. Authorities mentioned taking an essential authorization used for the whole API to increase extreme security in the mobile apps.

5.5. using of high-level authentication

Authentication machines stands the maximum vital portion of the mobile app security. As a designer besides a user authentication must be consider as more significant as of security view of opinion. One and only the most communal methods of authentication is done by PIN. So, PIN rule would be strong adequate thus, it cannot be cracked effortlessly. Multi factor authentication is one of the additional methods to brand your app further secure this can remain accomplished via the income of OTP login or authentication code scheduled in the mails then even further safety is thru biometrics [31].

5.6. Using a good cryptography tools and techniques

Important organization is a vital phase once it originates to encryption of your facts so make certain that you ensure not stiff central part of your encryption keys. Practice respectable procedures for encryption such as AES and SHA256 and always stock your secrets on limited devices. Practice the modern and reliable encryption methods [32].

5.7. Develop alter recognition techniques for your app

The technique remains to change to sirens when your code is actually changed or altered. Frequently it remains vital towards the record of code variations of your mobile app consequently that the malevolent computer programmer fix or insert the corrupted code in your app.

5.8. Deliver minimum rights

The principle of minimum rights is frequently needed for your app code security. It is desirable in the direction of stretch entree to the code to individual who remain proposed on the way to obtain them. Rest all are essentially not be specified the rights custody at least.

5.9. Test repetitively

An actual humble answer intended for the app is to examine constantly designed for the original variations as security characteristics remain altering gradually and thus you want to be essentially reorganized by the security tendencies acceptable to defend your app. You must choose for saturation testing and emulators to get a clue around the susceptibilities in your mobile app so, that they can remain more condensed [33].

6. NUMERICAL ANALYSIS

The below Table 1 contains the software critical success factors experienced in small and medium scale industry and the factors that are responsible for the software security aspects for IOS mobile application. The software critical success factors of small-medium scale industry (N) are Human recourse management, change in management, software developer involvement, software supplier network and relationship, software skill and expertise, software top management commitment, software performance monitoring, financial capabilities, software process management.

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Software critical success factor</th>
<th>Factors</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Human Resource Management (C1)</td>
<td>Write a Secure Code</td>
<td>Problem can be ducked if designers identify how to inscribe the code</td>
</tr>
<tr>
<td>2</td>
<td>Change in Management (C2)</td>
<td>Encryption of the data</td>
<td>Change an app in a popular way that all the facts involved in the app is coded actually fine, this is the unique practices.</td>
</tr>
<tr>
<td>3</td>
<td>Software developer Involvement (C3)</td>
<td>While using libraries be careful</td>
<td>The defects in the library can permit the invaders to practice malevolent code and bang the scheme.</td>
</tr>
<tr>
<td>4</td>
<td>Software supplier Network and relationship (C4)</td>
<td>Use authorized API</td>
<td>Authorities mention having an essential authorization for the complete API to increase extreme safety in the mobile applications Practical testing is done with combination</td>
</tr>
</tbody>
</table>

Table 1. Identification of Factors and their Characteristics for the software security issue while developing mobile application
Table 1. The factors of IOS mobile applications (M) are Write a secure code, Encryption of the data, while using libraries be careful, use authorized API, using a high-level authentication, using a good cryptography tools and techniques, develop alter recognition technique for your app, deliver minimum rights, test repetitively.

Table 2. AHP scale of Importance for Comparison pair and corresponding fuzzy numbers

Table 3. Pairwise comparison of the factors and sub-factors for the software security issues while developing the mobile application.

Table 4. Normalized Square Values of the level 1 Matrix
Develop an authentication technique for your App
Using a good cryptography tools and Techniques

Table 5. Distributive Normalization of matrix is obtained by original matrix divided by square root of the sum

<table>
<thead>
<tr>
<th>Alternatives/criteria</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write a Secure code</td>
<td>0.8926</td>
<td>0.9857</td>
<td>0.4489</td>
<td>0.0281</td>
<td>0.9810</td>
<td>0.7916</td>
<td>0.4764</td>
<td>0.4567</td>
<td>0.7174</td>
</tr>
<tr>
<td>Encryption of the data</td>
<td>0.0992</td>
<td>0.1095</td>
<td>0.3491</td>
<td>0.0321</td>
<td>0.1226</td>
<td>0.5277</td>
<td>0.2858</td>
<td>0.4567</td>
<td>0.4835</td>
</tr>
<tr>
<td>While using libraries be careful</td>
<td>0.0992</td>
<td>0.0357</td>
<td>0.4999</td>
<td>0.0046</td>
<td>0.0153</td>
<td>0.0189</td>
<td>0.0238</td>
<td>0.0254</td>
<td>0.0179</td>
</tr>
<tr>
<td>Use authorized API</td>
<td>0.1276</td>
<td>0.0137</td>
<td>0.4489</td>
<td>0.0040</td>
<td>0.0175</td>
<td>0.0220</td>
<td>0.0317</td>
<td>0.0381</td>
<td>0.0379</td>
</tr>
<tr>
<td>Using of high-level authentication</td>
<td>0.1116</td>
<td>0.1095</td>
<td>0.3990</td>
<td>0.0281</td>
<td>0.1226</td>
<td>0.2639</td>
<td>0.2858</td>
<td>0.0689</td>
<td>0.4835</td>
</tr>
<tr>
<td>Using a good cryptography tools and Techniques</td>
<td>0.1491</td>
<td>0.0274</td>
<td>0.3491</td>
<td>0.0241</td>
<td>0.0613</td>
<td>0.1319</td>
<td>0.4764</td>
<td>0.3045</td>
<td>0.2690</td>
</tr>
<tr>
<td>Develop alter recognition technique for your App</td>
<td>0.1785</td>
<td>0.0365</td>
<td>0.1995</td>
<td>0.0120</td>
<td>0.0408</td>
<td>0.0264</td>
<td>0.0953</td>
<td>0.0381</td>
<td>0.0179</td>
</tr>
<tr>
<td>Deliver minimum rights</td>
<td>0.2972</td>
<td>0.0565</td>
<td>0.2393</td>
<td>0.0164</td>
<td>0.0307</td>
<td>0.0660</td>
<td>0.3811</td>
<td>0.1522</td>
<td>0.0448</td>
</tr>
<tr>
<td>Test repetitively</td>
<td>0.1116</td>
<td>0.0219</td>
<td>0.2494</td>
<td>0.0201</td>
<td>0.0245</td>
<td>0.0439</td>
<td>0.4764</td>
<td>0.3045</td>
<td>0.0897</td>
</tr>
</tbody>
</table>

Table 6. Weighted Matrix

<table>
<thead>
<tr>
<th>Alternatives/criteria</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write a Secure code</td>
<td>0.3888</td>
<td>0.1544</td>
<td>0.0060</td>
<td>0.0007</td>
<td>0.1488</td>
<td>0.0817</td>
<td>0.0177</td>
<td>0.0313</td>
<td>0.0568</td>
</tr>
<tr>
<td>Encryption of the data</td>
<td>0.0376</td>
<td>0.0172</td>
<td>0.0047</td>
<td>0.1000</td>
<td>0.0186</td>
<td>0.0545</td>
<td>0.0016</td>
<td>0.0313</td>
<td>0.0292</td>
</tr>
<tr>
<td>While using libraries be careful</td>
<td>0.0376</td>
<td>0.0025</td>
<td>0.0007</td>
<td>0.0000</td>
<td>0.0023</td>
<td>0.0023</td>
<td>0.0112</td>
<td>0.0007</td>
<td>0.0012</td>
</tr>
<tr>
<td>Use authorized API</td>
<td>0.0485</td>
<td>0.0021</td>
<td>0.0007</td>
<td>0.1000</td>
<td>0.0186</td>
<td>0.0222</td>
<td>0.0106</td>
<td>0.0418</td>
<td>0.0292</td>
</tr>
<tr>
<td>Using of high-level authentication</td>
<td>0.0666</td>
<td>0.0043</td>
<td>0.0047</td>
<td>0.0006</td>
<td>0.0931</td>
<td>0.0316</td>
<td>0.0177</td>
<td>0.0209</td>
<td>0.0175</td>
</tr>
<tr>
<td>Using a good cryptography tools and Techniques</td>
<td>0.0678</td>
<td>0.0057</td>
<td>0.0027</td>
<td>0.0003</td>
<td>0.0662</td>
<td>0.0027</td>
<td>0.0035</td>
<td>0.0026</td>
<td>0.0012</td>
</tr>
<tr>
<td>Develop alter recognition technique for your App</td>
<td>0.1128</td>
<td>0.0057</td>
<td>0.0040</td>
<td>0.0004</td>
<td>0.0447</td>
<td>0.0068</td>
<td>0.0142</td>
<td>0.0104</td>
<td>0.0029</td>
</tr>
<tr>
<td>Deliver minimum rights</td>
<td>0.0424</td>
<td>0.0034</td>
<td>0.0033</td>
<td>0.0005</td>
<td>0.0337</td>
<td>0.0045</td>
<td>0.0177</td>
<td>0.0209</td>
<td>0.0058</td>
</tr>
</tbody>
</table>

Table 7. Normalized Weighted Matrix (Normalized Matrix * Weighted Matrix)

<table>
<thead>
<tr>
<th>Alternatives/criteria</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write a Secure code</td>
<td>0.3388</td>
<td>0.1544</td>
<td>0.0060</td>
<td>0.0007</td>
<td>0.1488</td>
<td>0.0817</td>
<td>0.0177</td>
<td>0.0313</td>
<td>0.0568</td>
</tr>
<tr>
<td>Encryption of the data</td>
<td>0.0376</td>
<td>0.0172</td>
<td>0.0047</td>
<td>0.1000</td>
<td>0.0186</td>
<td>0.0545</td>
<td>0.0016</td>
<td>0.0313</td>
<td>0.0292</td>
</tr>
<tr>
<td>While using libraries be careful</td>
<td>0.0376</td>
<td>0.0025</td>
<td>0.0007</td>
<td>0.0000</td>
<td>0.0023</td>
<td>0.0023</td>
<td>0.0112</td>
<td>0.0007</td>
<td>0.0012</td>
</tr>
<tr>
<td>Use authorized API</td>
<td>0.0485</td>
<td>0.0021</td>
<td>0.0007</td>
<td>0.1000</td>
<td>0.0186</td>
<td>0.0222</td>
<td>0.0106</td>
<td>0.0418</td>
<td>0.0292</td>
</tr>
<tr>
<td>Using of high-level authentication</td>
<td>0.0666</td>
<td>0.0043</td>
<td>0.0047</td>
<td>0.0006</td>
<td>0.0931</td>
<td>0.0316</td>
<td>0.0177</td>
<td>0.0209</td>
<td>0.0175</td>
</tr>
<tr>
<td>Using a good cryptography tools and Techniques</td>
<td>0.0678</td>
<td>0.0057</td>
<td>0.0027</td>
<td>0.0003</td>
<td>0.0662</td>
<td>0.0027</td>
<td>0.0035</td>
<td>0.0026</td>
<td>0.0012</td>
</tr>
<tr>
<td>Develop alter recognition technique for your App</td>
<td>0.1128</td>
<td>0.0057</td>
<td>0.0040</td>
<td>0.0004</td>
<td>0.0447</td>
<td>0.0068</td>
<td>0.0142</td>
<td>0.0104</td>
<td>0.0029</td>
</tr>
<tr>
<td>Deliver minimum rights</td>
<td>0.0424</td>
<td>0.0034</td>
<td>0.0033</td>
<td>0.0005</td>
<td>0.0337</td>
<td>0.0045</td>
<td>0.0177</td>
<td>0.0209</td>
<td>0.0058</td>
</tr>
<tr>
<td>Test repetitively</td>
<td>0.3388</td>
<td>0.1544</td>
<td>0.0060</td>
<td>0.0008</td>
<td>0.1488</td>
<td>0.0817</td>
<td>0.0177</td>
<td>0.0418</td>
<td>0.0012</td>
</tr>
<tr>
<td>Negative Solution</td>
<td>0.0376</td>
<td>0.0021</td>
<td>0.0007</td>
<td>0.0000</td>
<td>0.0023</td>
<td>0.0019</td>
<td>0.0009</td>
<td>0.0017</td>
<td>0.0568</td>
</tr>
</tbody>
</table>

Table 8. Compute the Euclidean distance amongst the goal alternate and the best/worst alternate

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>S1+</th>
<th>S1-</th>
<th>Relative closeness</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write a Secure code</td>
<td>0.0222</td>
<td>0.3334</td>
<td>0.9998</td>
<td>1</td>
</tr>
<tr>
<td>Encryption of the data</td>
<td>0.0043</td>
<td>0.0432</td>
<td>0.9100</td>
<td>2</td>
</tr>
<tr>
<td>While using libraries be careful</td>
<td>0.0377</td>
<td>0.0377</td>
<td>0.5000</td>
<td>4</td>
</tr>
<tr>
<td>Use authorized API</td>
<td>0.0485</td>
<td>0.0463</td>
<td>0.4885</td>
<td>7</td>
</tr>
<tr>
<td>Using of high-level authentication</td>
<td>0.0439</td>
<td>0.0422</td>
<td>0.4899</td>
<td>6</td>
</tr>
<tr>
<td>Using a good cryptography tools and Techniques</td>
<td>0.0536</td>
<td>0.0520</td>
<td>0.2747</td>
<td>9</td>
</tr>
<tr>
<td>Develop alter recognition technique for your App</td>
<td>0.0302</td>
<td>0.0194</td>
<td>0.3909</td>
<td>8</td>
</tr>
<tr>
<td>Deliver minimum rights</td>
<td>0.0647</td>
<td>0.0693</td>
<td>0.5171</td>
<td>3</td>
</tr>
<tr>
<td>Test repetitively</td>
<td>0.3413</td>
<td>0.3288</td>
<td>0.4907</td>
<td>5</td>
</tr>
</tbody>
</table>

Generate a matrix containing of M substitutes and N conditions. This matrix is typically named an “assessment matrix”. From the Table 1, M substitutes is nothing but the Alternatives which are defined. N condition is nothing but the criteria which are defined in the Table 3. All the criteria or conditions are multiplied.
together and taken the power 1/9 for each criterion. After that all the criteria are added together to get the resultant sum as 13.9236.

The Table 4 will show how to normalize the matrix each alternative is divided by the sum obtained earlier, that should be equal to 1. Each metric $j$ for each company $i$ is normalized to be in between 0 and 1. Each alternative has been squared to get the Table 4 matrix. All the criteria are added to get the sum of all the alternatives and the square root of the alternatives are taken as mentioned in the Table 4.

After calculating the sum and the square root of the sum we have to calculate the normalized matrix by doing the original matrix which is obtained in the Table 4 should be divided by square root of sum which is shown in the Table 5. Each criterion has been taken and it has to be divided by the sum of all the criteria their we will get the weights of the matrix which has been mentioned below the Table 4.

The higher its value the better the metric. From the Table 4. Write a secure code is 0.396, Encryption of the data 0.1566, while using libraries be careful 0.0134, use authorized API 0.0246, Using of high level authentication is 0.1517, using a good cryptography tools and techniques is 0.1032, Develop alter recognition technique for your App is 0.0372, Deliver minimum rights is 0.0686, Test repetitively is 0.0652. Compute the weighted normalized resultant matrix.

After calculating the sum and the square root of the sum we have to calculate the normalized matrix by doing the original matrix which is obtained in the Table 4 should be divided by square root of sum which is shown in the Table 5. Each criterion has been taken and it has to be divided by the sum of all the criteria their we will get the weights of the matrix which has been mentioned below the Table 4.

The higher its value the better the metric. From the Table 4. Write a secure code is 0.396, Encryption of the data 0.1566, while using libraries be careful 0.0134, use authorized API is 0.0246, Using of high level authentication is 0.1517, using a good cryptography tools and techniques is 0.1032, Develop alter recognition technique for your App is 0.0372, Deliver minimum rights is 0.0686, Test repetitively is 0.0652. Compute the weighted normalized resultant matrix.

The Table 6. It is imperative to message that individual standard must take its individual weight so that all of them will sum. The weights can be derivative arbitrarily (not recommended) or based on skilled information (industry standard). Normalized weighted matrix to be computed by multiplying the original matrix Table 5 with the weighted matrix Table 6 will give the resultant matrix which is obtained in the Table 7.

From the Table 7 we have to calculate the Positive ideal solution and Negative ideal solution for that we have to know that the non-functional and the functional factors which are defined in the alternatives.

For that we have to find the Euclidian distance for the best and the worst case of the alternatives or criteria defined in the Table 8. We want to find the maximum and minimum value of each economic metric amongst complete firms. This is an intention of the symmetrical detachment amongst the price of individual economic metric for a certain firm $i$ and the best/worst worth of such a metric amongst all firms.

For each alternate compute the comparison to the poorest alternate. The grades are our TOPSIS marks. Rank substitutes giving to the TOPSIS marks by descending order. The firm with metrics nearby to the finest will attain the maximum marks and consequently will be at the top of our ranking [34].

7. RESULTS COMPARISON STUDY WITH AHP AND TOPSIS

Policymaking is an extremely investigated subject and numerous approaches have been established to smoothen a policy-maker (PM) in selecting the top substitute [11]. Technique for order of preference by similarity to ideal solution (TOPSIS) is additional multi-criteria decision-making (MCDM) technique recognized by Hwang and Yoon in 2012 as a grade process [12]. This study is intensive on classifying which is the MCDM technique amongst AHP and TOPSIS. Subsequently TOPSIS is a grade technique, the writers suggest to associate AHP and TOPSIS techniques and regulate which technique's grade

Table 9. Performance of software security tools and production

<table>
<thead>
<tr>
<th>Small and medium scale Industry</th>
<th>Software security tools and machinery sector</th>
<th>Proposed software security tools and machinery sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK holds production</td>
<td>65%</td>
<td>85%</td>
</tr>
<tr>
<td></td>
<td>16%</td>
<td>25%</td>
</tr>
</tbody>
</table>
tools and machinery, the construction process must be carried out in high quality. Although techniques such as Authentication, secure code and cryptography are the main factors for software security. Though corporations have struggling in emerging secure products, enhancing manufacture process and consuming product practice information for their products, six-sigma methodology enables to overcome these difficulties [7]. The quantitative results achieved by Fuzzy AHP-TOPSIS will support the practitioners in categorizing higher ranked factors of 85% software-security-design while developing web application as shown in Table 9, so that the developers design an application whose security is supportable for a long time. Development guidelines can be produced over this estimation to help the developers in improving the design of supportable- security using high prioritized factors in concern [18]. Information composed from products, logistics operations and production processes help to improve 20% products and services as shown in Table 9 [21]. The fact that companies are ahead of the competition stems from the basic components of Android mobile application for software secure. Though, without the high and complete Software secure management system in the use of new technologies, they will not be able to have the necessary qualifications for software security for success [35].

8. CONCLUSION AND FURTHER ENHANCEMENT

In small and medium scale software business are directing the software problems in software improvement process in software serious achievement aspects. Approximately the software security difficulties in the six-sigma and fuzzy logic using different methodology were discussed by the different authors. A case study of mobile application for small and medium scale industry efficiency of using this method is demonstrated from the theoretic point of interpretation.. Nine components were taken for the experiments and were designed to verify the software security factors in mobile application performance of ranking by using the AHP and TOPSIS method. The experiment result shows the Ranking of the factors in the software security mobile application. In this way, many tasks will compete for the same resources. However, this method shows advantages of getting the ranking for software security in mobile application using the TOPSIS Method. These are nearly the finest performs that a mobile application designer essentially tracks in demand to consume a completely protected problematic to crash the application. In the current year cyber security has confirmed its reputation and customers are nowadays attracted in extra protected application to trust upon where the out of scope of this work, this limitation needs to be addressed in the upcoming research work.

FUTURE ENHANCEMENT

In the nearby upcoming days safety or security resolve the performance as unique distinguishing and challenging in the application ecosystem with clients. The choosing safety application to preserve secrecy of their information over additional mobile application.

REFERENCES


[8] Seyhan, Kübra, Tu N. Nguyen, Sedat Akleylek, Korhan Cengiz, and SK Hafizul Islam. "Bi-GISIS KE: Modified key exchange protocol with reusable


[32] Lorint R, Marcu M. Survey regarding the way students perceive security permissions of mobile applications. In 2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI) 2019 May 29 (pp. 000287-000290). IEEE.

