Atlantis Highlights in Computer Sciences, volume 4
Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)

Critical Success Factor Estimation for Software Security in Small and Medium Scale Industry Using AHP and TOPSIS Approach

Kavyashree N[1,*], Supriya M C[2], Lokesh M R[3]

[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.

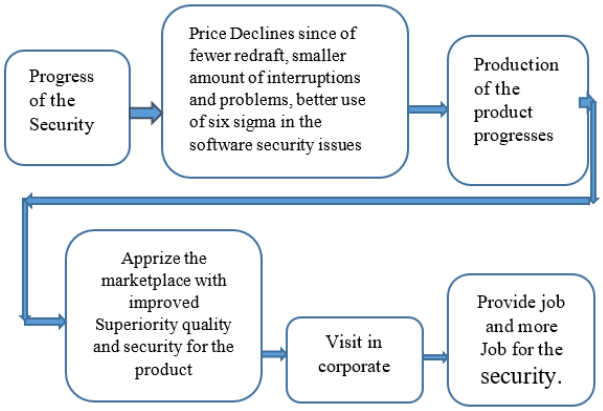

Figure 1 Importance of six sigma in software security for small and medium scale industry

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 days are underneath occurrence from outside intimidations and facts threats. Figure-2 Shows the absence of complexity round 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 fight contrary 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**.

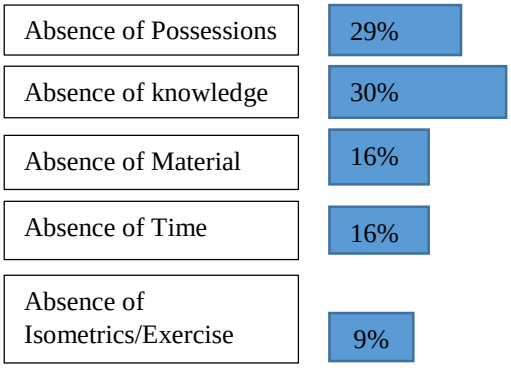

Figure 2 Draw back in small and medium scale industry

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 vehicles [11]. Development of TOPSIS Technique under 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

integrating maximizing deviation and TOPSIS methods [13]. A novel Pythagorean fuzzy AHP and fuzzy TOPSIS methodology for green supplier selection in the Industry 4.0 era [14]. An Integrated Fuzzy Best-Worst-TOPSIS Method for Evaluation of Hotel Website and Digital Solutions Provider Firms [15]. TOPSIS method for developing supplier selection with probabilistic linguistic information [16].

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.

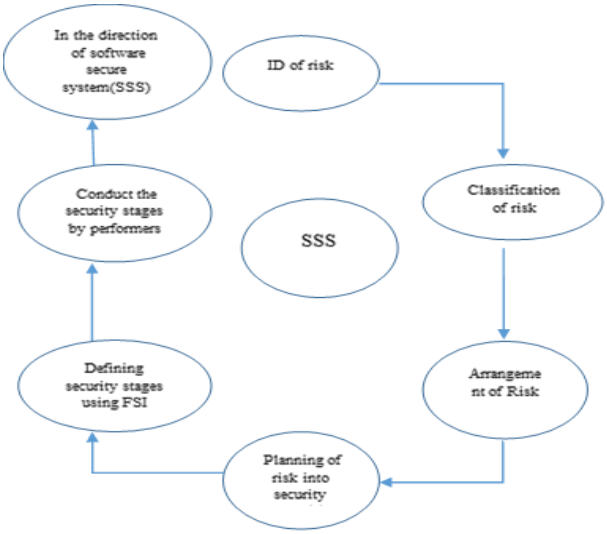

Figure 3 Several phases of SSS

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 flattering 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).

$$(a_{ij})_{M\times N} \tag{1}$$

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

$$\alpha_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{M}(a_{ij})^2}} \tag{2}$$

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).

$$\chi_{ij} = \alpha_{ij} * \omega_j \quad (3)$$

$$\omega_j = \frac{w_j}{\sum_{j=1}^{N} w_j} \quad (4)$$

$$\sum_{j=1}^{N} \omega_j = 1 \quad (5)$$

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 an each normalized value obtained from step 2.

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

$$\chi_j^b = \max_{i=1}^{M} \chi_{ij} \quad (6)$$

$$\chi_j^w = \min_{i=1}^{M} \chi_{ij} \quad (7)$$

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:

$$d_i^b = \sqrt{\sum_{j=1}^{N} (\chi_{ij} - \chi_j^b)^2} \quad (8)$$

$$d_i^w = \sqrt{\sum_{j=1}^{N} (\chi_{ij} - \chi_j^w)^2} \quad (9)$$

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

$$s_i = \frac{d_i^w}{d_i^w + d_i^b} \quad (10)$$

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 1. Identification of Factors and their Characteristics for the software security issue while developing mobile application

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

5	Software skill and expertise (C5)	Using of high-level authentication	Multi factor authentication is one of the additional methods to brand your application are more secure this can be accomplished via the income of OTP login or authentication code on mails and level further secure is done by biometrics
6	Software Top Management Commitment (C6)	Using a good cryptography tools and techniques	Use good protocols for encryption, always stock your solutions on limited devices. Practice the modern and reliable encryption methods
7	Software Performance Monitoring (C7)	Develop alter recognition techniques for your app	On the way to get sirens once your code is actually changed or altered. To initiate the strategic at your application, keep log of activities.
8	Financial Capabilities (C8)	Deliver minimum rights	The principle of minimum rights is frequently desirable to stretch access to the code to individual those who are proposed to obtain them
9	Software Process Management (C9)	Test repetitively	security characteristics are altering gradually and so you need to be essentially reorganized with the security tendencies acceptable to defend your application.

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.

The Table 1 defines the details information about the software critical success factor and factors that are responsible for the development of software security of IOS mobile application and characteristics will explain the information about what the factors responsible for the IOS mobile. The Table 2 AHP scale of Importance for Comparison pair and corresponding fuzzy numbers which is very important for the comparison of the software critical success factors and the factors responsible for the development of the software security for mobile application in the IOS.

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

AHP Scale of Importance for comparison pair (aij)	Numeric Rating	Reciprocal (decimal)
Extreme Importance	9	1/9 (0.111)
Very strong to extremely	8	1/8 (0.125)
Very strong Importance	7	1/7 (0.143)
Strongly to to very strong	6	1/6(0.167)
Strong Importance	5	1/5(0.200)
Moderately to Strong	4	1/4(0.250)
Moderate Importance	3	1/3(0.333)
Equally to Moderately	2	1/2(0.500)
Equal Importance	1	1 (1.000)

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

Alternatives/criteria	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**Summation of alternatives**	**weight**
Write a Secure code	1.0000	9.0000	9.0000	7.0000	8.0000	6.0000	5.0000	3.0000	8.0000	5.2851	0.3796
Encryption of the data	0.1111	1.0000	7.0000	8.0000	1.0000	4.0000	3.0000	3.0000	5.0000	2.1800	0.1566
While using libraries be careful	0.1111	0.1433	1.0000	0.1111	0.1250	0.1430	0.2500	0.1670	0.2000	0.1861	0.0134
Use authorized API	0.1430	0.1250	9.0000	1.0000	0.1430	0.1670	0.3330	0.2500	0.2000	0.3423	0.0246
Using of high-level authentication	0.1250	1.0000	8.0000	7.0000	1.0000	2.0000	3.0000	4.0000	5.0000	2.1115	0.1517
Using a good cryptography tools and Techniques	0.1670	0.2500	7.0000	6.0000	0.5000	1.0000	5.0000	2.0000	3.0000	1.4375	0.10372
Develop alter recognition technique for your App	0.2000	0.3330	4.0000	3.0000	0.3330	0.2000	1.0000	0.2500	0.2000	0.5178	0.0372
Deliver minimum rights	0.3330	0.3330	6.0000	4.0000	0.2500	0.5000	4.0000	1.0000	0.5000	0.9558	0.0686
Test repetitively	0.1250	0.2000	5.0000	5.0000	0.2000	0.3330	5.0000	2.0000	1.0000	0.9073	0.0652
SUM										13.9236	

Table 4. Normalized Square Values of the level 1 Matrix

Alternatives/criteria	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
Write a Secure code	1.0000	81.0000	81.0000	49.0000	64.0000	36.0000	25.0000	9.0000	64.0000
Encryption of the data	0.0123	1.0000	49.0000	64.0000	1.0000	16.0000	9.0000	9.0000	25.0000
While using libraries be careful	0.0123	0.0205	1.0000	0.0123	0.0156	0.0204	0.0625	0.0279	0.0400
Use authorized API	0.0204	0.0156	81.0000	1.0000	0.0204	0.0279	0.1109	0.0625	0.0400

Using of high-level authentication	0.0156	1.0000	64.0000	49.0000	1.0000	4.0000	9.0000	16.0000	25.0000
Using a good cryptography tools and Techniques	0.0279	0.0625	49.0000	36.0000	0.2500	1.0000	25.0000	4.0000	9.0000
Develop alter recognition technique for your App	0.0400	0.1109	16.0000	9.0000	0.1109	0.0400	1.0000	0.0625	0.0400
Deliver minimum rights	0.1109	0.1109	36.0000	16.0000	0.0625	0.2500	16.0000	1.0000	0.2500
Test repetitively	0.0156	0.0400	25.0000	25.0000	0.0400	0.1109	25.0000	4.0000	1.0000
SUM	1.2552	83.3604	402.0000	249.0123	66.4995	57.4492	110.1734	43.1529	124.3700
Square root of sum	1.120340761	9.130194	20.04994	15.78012	8.15472	7.579527	10.49635	6.569086	11.15213

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

Alternatives/criteria	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
Write a Secure code	0.8926	0.9857	0.4489	0.0281	0.9810	0.7916	0.4764	0.4567	0.7174
Encryption of the data	0.0992	0.1095	0.3491	0.0321	0.1226	0.5277	0.2858	0.4567	0.4483
While using libraries be careful	0.0992	0.0157	0.0499	0.0004	0.0153	0.0189	0.0238	0.0254	0.0179
Use authorized API	0.1276	0.0137	0.4489	0.0040	0.0175	0.0220	0.0317	0.0381	0.0179
Using of high-level authentication	0.1116	0.1095	0.3990	0.0281	0.1226	0.2639	0.2858	0.6089	0.4483
Using a good cryptography tools and Techniques	0.1491	0.0274	0.3491	0.0241	0.0613	0.1319	0.4764	0.3045	0.2690
Develop alter recognition technique for your App	0.1785	0.0365	0.1995	0.0120	0.0408	0.0264	0.0953	0.0381	0.0179
Deliver minimum rights	0.2972	0.0365	0.2993	0.0161	0.0307	0.0660	0.3811	0.1522	0.0448
Test repetitively	0.1116	0.0219	0.2494	0.0201	0.0245	0.0439	0.4764	0.3045	0.0897

Table 6. Weighted Matrix

Alternatives/criteria	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
Write a Secure code	0.3388	0.1544	0.0060	0.0007	0.1488	0.0817	0.0177	0.0313	0.0468
Encryption of the data	0.0376	0.0172	0.0047	0.0008	0.0186	0.0545	0.0106	0.0313	0.0292
While using libraries be careful	0.0376	0.0025	0.0007	0.0000	0.0023	0.0019	0.0009	0.0017	0.0012
Use authorized API	0.0485	0.0021	0.0060	0.0001	0.0027	0.0023	0.0012	0.0026	0.0012
Using of high-level authentication	0.0424	0.0172	0.0053	0.0007	0.0186	0.0272	0.0106	0.0418	0.0292
Using a good cryptography tools and Techniques	0.0566	0.0043	0.0047	0.0006	0.0093	0.0136	0.0177	0.0209	0.0175
Develop alter recognition technique for your App	0.0678	0.0057	0.0027	0.0003	0.0062	0.0027	0.0035	0.0026	0.0012
Deliver minimum rights	0.1128	0.0057	0.0040	0.0004	0.0047	0.0068	0.0142	0.0104	0.0029
Test repetitively	0.0424	0.0034	0.0033	0.0005	0.0037	0.0045	0.0177	0.0209	0.0058

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

Alternatives/criteria	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**
Write a Secure code	0.3388	0.1544	0.0060	0.0007	0.1488	0.0817	0.0177	0.0313	0.0468
Encryption of the data	0.0376	0.0172	0.0047	0.0008	0.0186	0.0545	0.0106	0.0313	0.0292
While using libraries be careful	0.0376	0.0025	0.0007	0.0000	0.0023	0.0019	0.0009	0.0017	0.0012
Use authorized API	0.0485	0.0021	0.0060	0.0001	0.0027	0.0023	0.0012	0.0026	0.0012
Using of high-level authentication	0.0424	0.0172	0.0053	0.0007	0.0186	0.0272	0.0106	0.0418	0.0292
Using a good cryptography tools and Techniques	0.0566	0.0043	0.0047	0.0006	0.0093	0.0136	0.0177	0.0209	0.0175
Develop alter recognition technique for your App	0.0678	0.0057	0.0027	0.0003	0.0062	0.0027	0.0035	0.0026	0.0012
Deliver minimum rights	0.1128	0.0057	0.0040	0.0004	0.0047	0.0068	0.0142	0.0104	0.0029
Test repetitively	0.0424	0.0034	0.0033	0.0005	0.0037	0.0045	0.0177	0.0209	0.0058
Positive Solution	0.3388	0.1544	0.0060	0.0008	0.1488	0.0817	0.0177	0.0418	0.0012
Negative Solution	0.0376	0.0021	0.0007	0.0000	0.0023	0.0019	0.0009	0.0017	0.0468

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

Alternatives	**S1+**	**S1-**	**Relative closeness**	**Rank**
Write a Secure code	0.0022	0.3534	0.9938	1
Encryption of the data	0.0043	0.0432	0.9100	2
While using libraries be careful	0.0377	0.0377	0.5000	4
Use authorized API	0.0485	0.0463	0.4885	7
Using of high-level authentication	0.0439	0.0422	0.4899	6
Using a good cryptography tools and Techniques	0.0536	0.0203	0.2747	9
Develop alter recognition technique for your App	0.0302	0.0194	0.3909	8
Deliver minimum rights	0.0647	0.0693	0.5171	3
Test repetitively	0.3413	0.3288	0.4907	5

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 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.

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 TOPSIS combination, and TOPSIS with equal weights) bring into line additional carefully with the DM's original predilection [14].

Table 9. Performance of software security tools and production

Small and medium scale Industry	**software security tools and machinery sector**	**Proposed software security tools and machinery sector**
UK holds	65%	85%
production	16%	25%

Decision makers need to rise their existence in the world market by contributing the Software security products. Thus, the criteria yield determined weightage by decision-makers. UK holds over 65% of the software security tools and machinery sector. London which is one of the important centers of internal and external markets [5]. In direction to rise the competence of the production that keeps with 16% of the most formed Software security

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 FUTHER 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 ecosphere with clients. The choosing safety application to preserve secrecy of their information over additional mobile application.

REFERENCES

[1] Zarour, Mohammad, Mamdouh Alenezi, and Khalid Alsarayrah. "Software Security Specifications and Design: How Software Engineers and Practitioners Are Mixing Things up." In Proceedings of the Evaluation and Assessment in Software Engineering, pp. 451-456. 2020. https://doi.org/10.1016/j.infsof.2020.106488

[2] Mirakhorli, Mehdi, Matthias Galster, and Laurie Williams. "Understanding Software Security from Design to Deployment." ACM SIGSOFT Software Engineering Notes 45, no. 2 (2020): 25-26. https://doi.org/10.1145/3385678.3385687

[3] Wen, Shao-Fang, and Basel Katt. "Learning Software Security in Context: An Evaluation in Open Source Software Development Environment." In Proceedings of the 14th International Conference on Availability, Reliability and Security, pp. 1-10. 2019. https://doi.org/10.1145/3339252.3340336

[4] Galhardo, Carlos Cardoso, Peter Mell, Irena Bojanova, and Assane Gueye. "Measurements of the Most Significant Software Security Weaknesses." In Annual Computer Security Applications Conference, pp. 154-164. 2020. https://doi.org/10.1145/3427228.3427257

[5] Venson, Elaine. "The effects of required security on software development effort." In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings, pp. 166-169. 2020. https://doi.org/10.1145/3377812.3381393

[6] Yang H, Rao P, Simpson T, Lu Y, Witherell P, Nassar AR, Reutzel E, Kumara S. Six-Sigma Quality Management of Additive Manufacturing. Proceedings of the IEEE. 2020 Nov 26.

[7] Hu, Liwen, Ngoc-Tu Nguyen, Wenjin Tao, Ming C. Leu, Xiaoqing Frank Liu, Md Rakib Shahriar, and SM Nahian Al Sunny. "Modeling of cloud-based digital twins for smart manufacturing with MT connect." Procedia manufacturing 26 (2018): 1193-1203.

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

keys for IoT security." Journal of Information Security and Applications 58 (2021): 102788.

[9] Nguyen, Tu N., Bing-Hong Liu, Nam P. Nguyen, and Jung-Te Chou. "Cyber security of smart grid: attacks and defenses." In ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1-6. IEEE, 2020.

[10] Veena TR, Prabhushankar GV. Roadmap for integration of Lean Six Sigma methodology with ISO 9001: 2015 QMS standard. International Journal of Advanced Operations Management. 2020;12(4):303-29.

[11] Galli BJ. Measurement System Analysis and System Thinking in Six Sigma: How They Relate and How to Use Them. International Journal of System Dynamics Applications (IJSDA). 2020 Jan 1;9(1):44-62.

[12] Sony M, Antony J, Park S, Mutingi M. Key criticisms of Six Sigma: a systematic literature review. IEEE Transactions on Engineering Management. 2019 Feb 8;67(3):950-62.

[13] Schäfer F, Schwulera E, Otten H, Franke J. From Descriptive to Predictive Six Sigma: Machine Learning for Predictive Maintenance. In2019 Second International Conference on Artificial Intelligence for Industries (AI4I) 2019 Sep 25 (pp. 35-38). IEEE.

[14] Arun, M., E. Baraneetharan, A. Kanchana, and S. Prabu. "Detection and monitoring of the asymptotic COVID-19 patients using IoT devices and sensors." International Journal of Pervasive Computing and Communications (2020).

[15] Kumar, M. Keerthi, B. D. Parameshachari, S. Prabu, and Silvia liberata Ullo. "Comparative Analysis to Identify Efficient Technique for Interfacing BCI System." In IOP Conference Series: Materials Science and Engineering, vol. 925, no. 1, p. 012062. IOP Publishing, 2020.

[16] Bakioglu G, Atahan AO. AHP integrated TOPSIS and VIKOR methods with Pythagorean fuzzy sets to prioritize risks in self-driving vehicles. Applied Soft Computing. 2021 Feb 1;99:106948. https://doi.org/10.1016/j.asoc.2020.106948

[17] Zulqarnain RM, Siddique I, Jarad F, Ali R, Abdeljawad T. Development of TOPSIS Technique under Pythagorean Fuzzy Hypersoft Environment Based on Correlation Coefficient and Its Application towards the Selection of Antivirus Mask in COVID-19 Pandemic Complexity. 2021 Mar 17;2021. https://doi.org/10.1155/2021/6634991

[18] Farrokhizadeh E, Seyfi-Shishavan SA, Gündoğdu FK, Donyatalab Y, Kahraman C, Seifi SH. A spherical fuzzy methodology integrating maximizing deviation and TOPSIS methods. Engineering Applications of Artificial Intelligence. 2021 May 1;101:104212. https://doi.org/10.1016/j.engappai.2021.104212

[19] Subramani, Prabu, Ganesh Babu Rajendran, Jewel Sengupta, Rocío Pérez de Prado, and Parameshachari Bidare Divakarachari. "A block bi-diagonalization-based pre-coding for indoor multiple-input-multiple-output-visible light communication system." Energies 13, no. 13 (2020): 3466.

[20] Rajendrakumar, Shiny, and V. K. Parvati. "Automation of irrigation system through embedded computing technology." In Proceedings of the 3rd International Conference on Cryptography, Security and Privacy, pp. 289-293. 2019.

[21] Kumar, M. Keerthi, B. D. Parameshachari, S. Prabu, and Silvia liberata Ullo. "Comparative Analysis to Identify Efficient Technique for Interfacing BCI System." In IOP Conference Series: Materials Science and Engineering, vol. 925, no. 1, p. 012062. IOP Publishing, 2020.

[22] L. Tan, K. Yu, F. Ming, X. Cheng, G. Srivastava, “Secure and Resilient Artificial Intelligence of Things: a HoneyNet Approach for Threat Detection and Situational Awareness”, IEEE Consumer Electronics Magazine, 2021, doi: 10.1109/MCE.2021.3081874.

[23] L. Tan, N. Shi, K. Yu, M. Aloqaily, Y. Jararweh, “A Blockchain-Empowered Access Control Framework for Smart Devices in Green Internet of Things”, ACM Transactions on Internet Technology, vol. 21, no. 3, pp. 1-20, 2021,https://doi.org/10.1145/3433542.

[24] Z. Guo, A. K. Bashir, K. Yu, J. C. Lin, Y. Shen, “Graph Embedding-based Intelligent Industrial Decision for Complex Sewage Treatment Processes”, International Journal of Intelligent Systems，2021, doi: 10.1002/int.22540.

[25] Çalık A. A novel Pythagorean fuzzy AHP and fuzzy TOPSIS methodology for green supplier selection in the Industry 4.0 era. Soft Computing. 2021 Feb;25(3):2253-65. https://doi.org/10.1007/s00500-020-05294-9

[26] Samanlioglu F, Burnaz AN, Diş B, Tabaş MD, Adıgüzel M. An Integrated Fuzzy Best-Worst-TOPSIS Method for Evaluation of Hotel Website and Digital Solutions Provider Firms. Advances in

Fuzzy Systems. 2020 Nov 3;2020. https://doi.org/10.1155/2020/8852223

[27] Lei F, Wei G, Gao H, Wu J, Wei C. TOPSIS method for developing supplier selection with probabilistic linguistic information. International Journal of Fuzzy Systems. 2020 Mar 9:1-1. https://doi.org/10.1007/s40815-019-00797-6

[28] Balapour A, Nikkhah HR, Sabherwal R. Mobile application security: Role of perceived privacy as the predictor of security perceptions. International Journal of Information Management. 2020 Jun 1;52:102063.

[29] Sykes ER. A context-aware system using mobile applications and beacons for on-premise security environments. Journal of Ambient Intelligence and Humanized Computing. 2020 Nov;11(11):5487-511.

[30] Magklaras G, López-Bojórquez LN. A review of information security aspects of the emerging COVID-19 contact tracing mobile phone applications. In International Symposium on Human Aspects of Information Security and Assurance 2020 Jul 8 (pp. 30-44). Springer, Cham.

[31] Palma F, Realista N, Serrão C, Nunes L, Oliveira J, Almeida A. Automated security testing of Android applications for secure mobile development. In2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW) 2020 Oct 24 (pp. 222-231). IEEE.

[32] Lorint R, Marcu M. Survey regarding the way students perceive security permissions of mobile applications. In2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI) 2019 May 29 (pp. 000287-000290). IEEE

[33] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. Lin, T. Sato, “Secure Artificial Intelligence of Things for Implicit Group Recommendations”, IEEE Internet of Things Journal, 2021, doi: 10.1109/JIOT.2021.3079574.

[34] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin and G. Srivastava, "An Efficient Ciphertext-Policy Weighted Attribute-Based Encryption for the Internet of Health Things," IEEE Journal of Biomedical and Health Informatics, 2021, doi: 10.1109/JBHI.2021.3075995.

[35] L. Zhen, A. K. Bashir, K. Yu, Y. D. Al-Otaibi, C. H. Foh, and P. Xiao, “Energy-Efficient Random Access for LEO Satellite-Assisted 6G Internet of Remote Things”, IEEE Internet of Things Journal, doi: 10.1109/JIOT.2020.3030856.