ISSN 2347 - 3983
Volume 8. No. 5, May 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter26852020.pdf
https://doi.org/10.30534/ijeter/2020/26852020

Impact Cultural-Quality Factors on Successes and Failures Software System

Ayat Mohammad Erman[1], **Hamed Fawareh**[2]
[1]Zarqa University, Jordan, Ayaterman_se2@hotmai.com
[2]Zarqa University, Jordan, fawareh@zu.edu.jo

ABSTRACT

Researchers have often attempted to raise the success rate of software systems over the past century. Improve software quality models and other software elements to make them more customer satisfaction and achieve customer permanence. Several quality models and variables have proposed to decrease software system failure and complexity. Also, several quality models proposed to assess the general and particular types of software products. These models have proposed to determine the general or particular scopes of the software system. The proposed models are evaluated based on comparisons between the well-known models to customise the closed model. These comparisons are the leakage of criteria based on distinct views and knowledge of cultural and social habits. New factors proposed by the customise software quality models. In this paper, a cultural model proposed. This model based is on six criteria, namely: Natural Language, Religion, Social Habits, Custom, Ethics, and Law. In this model, the new criterions factors categorised into three clusters, each cluster containing the result of the proposed cultural model explaining that the six criteria factors should consider influencing the user's approval or abstention from the program, depending on its culture, which will reflect in the quality of the program system. Several basic rules have established for each cluster of culture factors, the main task of these rules is to help and assist in suggesting quality Equations for software systems in order to evaluate Natural Language, Religion, Social Habits, Customs, Ethics and law factors.

Key words: Software quality, quality model, culture quality factor, software matrics.

1. INTRODUCTION

The growing need for software products has increased competition in the software market, Software quality becomes more important, which plays an essential role in the overall software system's success; it considered an essential aspect for developers, users and managers of projects. ISO defines the quality of the software as a collection of characteristics that describe and assess the software product quality [1]. Quality assurance is a formal process to ensure the quality of the product and control the methods used to ensure quality [2]. In the literature of software engineering, there are many quality models; each model contains different quality characteristics or factors [1]. These models proposed to evaluate all software products by determining the attributes software product, enabling us to evaluate each attribute [3-4]. Considered the completion of software systems and met user expectations and business performance goals are not an easy job, although it has met all the specified features in a quality assurance model .

Four aspects have been identified based on recent studies [5], to assess the success or failure of the software system. These aspects are: (Implementation process, Recognize software value, Customer satisfaction and culture requirement)

Success is found relatively rare in the world of software projects. One potential reason might be the difference in the culture of the meaning of success in the minds of people evaluating the quality of the project. Therefore, the criteria for project success, as believed by various stakeholder groups, do not match. The highest determining factor of achievement is the functionality and quality of the project outcome, success in external goals such as customer satisfaction.

Cultural and social compatibility is essential to the acceptance of the software system. Spencer-Oatey and Franklin pointed out the culture associated with human existence provided in conjunction with the development of human life, according to individual creativity and production in various areas [6]. It is a vital aspect of software systems, as will discuss during cases studies.

Several software projects achieved great success because they met cultural factors such as Microsoft Dynamics AX and WeChat is used throughout the world globally because this software obtains a cultural sub-factor [7-8]. While Blue-Whale Challenge game considered as a failure software system, this game does not take into account the cultural-quality factor [9].

Due to the widespread of software systems, the quality of the software systems is considered very important. That is known as the system's ability to meet the expected technical requirements within structured and planned procedures.

Over the previous years, plenty of quality models of software systems were proposed, which contained many quality factors and criteria which will ensure a high-quality system for the user.

However, a recent study showed that the success rate of the software systems is relatively rare. One of the possible reasons is the difference in the culture of the meaning of success in people's opinions and those who use the software system. That is why the success standards of the software system are not identical, because they differ according to the concerned beliefs. In other words, some users might approve on some systems and other might reluctant to them according to their culture, language, religion or habits.

The concept of culture included a group of elements, some of which connected to the software system's quality, and some of which not connected to the software system's quality. This paper focused on the elements of culture that are related to the software system's quality and has a direct effect on the success or failure of the system.

This paper focused on how to develop new quality software system models in which this model takes into consideration the social and cultural elements of the user, therefore overcoming the problem of the user's non-acceptance for the software system because it does not fit their culture and decreases the software system's usability, therefore causes its failure.

The proposed model included quality criteria for the culture factor as well as the need to propose software criteria in order to get quantities values that help the concerned to evaluate the software system's quality from the cultural and social side.

2. QUALITY MODELS BACKGROUND

In the literature of software engineering, there are many quality models; each model contains different quality characteristics or factors [10]. These models have suggested evaluating general and specific types of software products [11].

McCall proposed the first model in 1977, which defines the qualities of the software product as a hierarchy of factors, criteria, and Equations. The factors describe the system characteristics, a quality criterion is an attribute of software production and design-related quality factor, and Equations defining and using a measurement scale and method [12].

This model contains eleven factors and twenty-three criteria; these factors divided into three groups of products: transition, revision, and operations. Because this model is ancient, there was no consideration for new features of systems such as security and social requirements; it has not taken into account unique characteristics of systems such as safety and social factors [13].

The second model called Boehm [13]. This model defined the primary quality characteristic as a general utility. The main purpose of this model is to address the contemporary weaknesses of models that evaluate software quality automatically and quantitatively. This model discussed the high-level characteristics and classified them into three groups: general utility as a utility, maintenance, and portability. Seven qualities collectively characteristics represent the qualities expected from a software system: portability, reliability, efficiency, usability, testability, comprehensibility, flexibility and human engineering [13]

The third model suggested by Dromey. He introduced a framework for assessing the requirements, designing, and implementation of the system. He indicates that the evaluation for each product is different, so we need a dynamic modelling idea. Therefore, the primary objective of the proposed model was to obtain a model that was broad enough for different systems to work [14].

The model aimed at enhancing understanding of the relationship between quality attributes (characteristics) and sub-attributes (sub-characteristics)- several attributes defined in this model, such as the layer, high-level attributes, and subordinate attributes. One of the main drawbacks of this model is that it suffers from a lack of software quality measurement criteria [14].

The fourth model, proposed by Robert and Hewlett-Packard called FURPS. In this model, the features classified into two categories according to the functional and non-functional requirements of the user [4].

- The functional requirements are the input and expected output is defined.
- Non-functional requirements are usability, reliability, performance, sustainability, and usability, which includes human factors, aesthetics, user documentation and material of training [5].

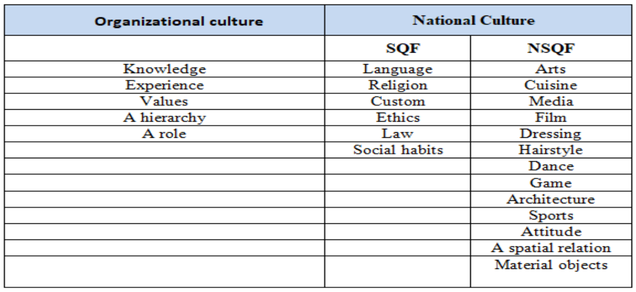
The fifth model proposed by ISO 9000 model, which considered the most basic standard for quality assurance. Total quality attributes of software products have classified as characteristics and sub-characteristics in a hierarchical tree structure. These characteristics include Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability, further divided into twenty-one sub-characteristics. The defined characteristics in this model can apply to all software types, including firmware computer programs and data, and it can provide consistent software product quality terminology. They also offer a framework for trade-offs between the capabilities of software products [16].

3. THE PROPOSED CULTURAL-QUALITY SOFTWARE MODEL

This paper proposed a software quality model in the presence of culture quality factors. The cultural-quality factor is essential in the quality software system. It plays the main rules in the success and failure of the software systems. We specify the main factor affecting the success and failure of software by analysing the definition of culture.

According to a literature review in sociology and anthropology, we have obtained several culture definitions as O’Reilly III, and other defines culture as "a way of life for different races/ ethnicity encompasses many facts like religions, languages, dressing attires, hairstyles/cuisines food eaten/ certain games/ sports/ martial arts practiced / certain musical played, certain songs/ music's dances, values systems." [17]. Based on the analysing of the cultural definitions, twenty-four elements extracted. We compare culture elements and remove the repetition of similar meaning. Twenty-two elements consider cultural factors. We classified the factors into two sets of organisation culture and national culture shown in table 1. we exclude organisational culture factors because of its impact on the production process of the software system. Furthermore, we consider the impact of national-cultural factors related to Software engineering Quality Factors (SQF).

Table 1:.Classification Culture Factors



Organizational culture	National Culture	
	SQF	NSQF
Knowledge	Language	Arts
Experience	Religion	Cuisine
Values	Custom	Media
A hierarchy	Ethics	Film
A role	Law	Dressing
	Social habits	Hairstyle
		Dance
		Game
		Architecture
		Sports
		Attitude
		A spatial relation
		Material objects

Criteria represent the central part of evaluating and defining any quality factor. These criteria may be characteristic of the software product or attributes of the production operation or process. By studying and analysing the definition of culture, and its impacts on Software engineering Quality Factors (SQF) we determined the culture element, that is related to Software engineering Quality Factors (SQF) as a set of criteria of culture quality factor, which are language, religion, customs, Social habits, Law and Ethics.

Based on the analysis of cultural elements and understanding of quality criteria, in this thesis, we considered the national-cultural factors which related to Software engineering Quality Factors (SQF) are the criteria of the cultural-quality factor which are language, religion, customs, Social habits, Law and Ethics.

We categorised these criteria into three main groups based on a study of their definition, to find a clear way to measure and evaluate these criteria.

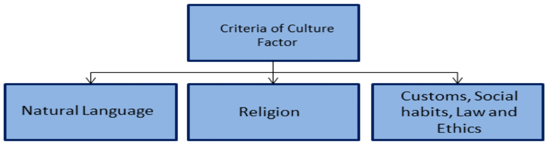
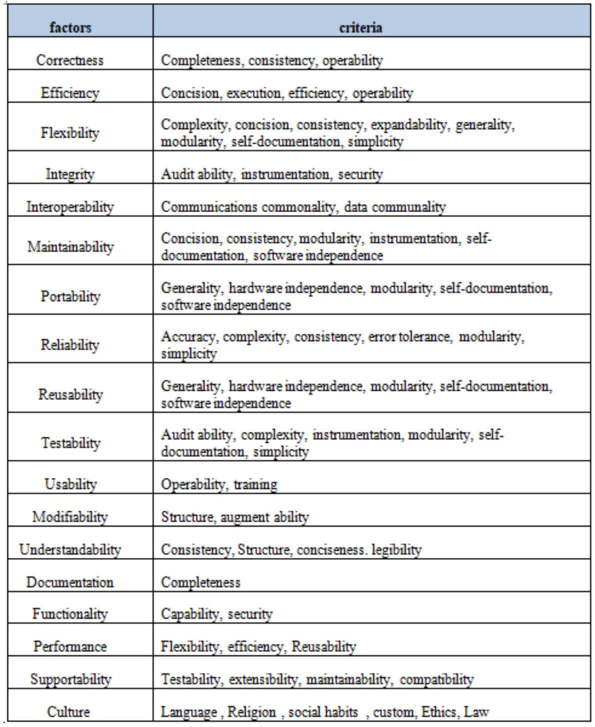

Criteria of Culture Factor
Natural Language
Religion
Customs, Social habits, Law and Ethics

Figure 1: Culture Criteria

Table 2: Quality Factor and criteria



factors	criteria
Correctness	Completeness, consistency, operability
Efficiency	Concision, execution, efficiency, operability
Flexibility	Complexity, concision, consistency, expandability, generality, modularity, self-documentation, simplicity
Integrity	Audit ability, instrumentation, security
Interoperability	Communications commonality, data communality
Maintainability	Concision, consistency, modularity, instrumentation, self-documentation, software independence
Portability	Generality, hardware independence, modularity, self-documentation, software independence
Reliability	Accuracy, complexity, consistency, error tolerance, modularity, simplicity
Reusability	Generality, hardware independence, modularity, self-documentation, software independence
Testability	Audit ability, complexity, instrumentation, modularity, self-documentation, simplicity
Usability	Operability, training
Modifiability	Structure, augment ability
Understandability	Consistency, Structure, conciseness. legibility
Documentation	Completeness
Functionality	Capability, security
Performance	Flexibility, efficiency, Reusability
Supportability	Testability, extensibility, maintainability, compatibility
Culture	Language , Religion , social habits , custom, Ethics, Law

3.1 Natural Factor

The natural language defined as a group of words that are combined to form sentences and are used by people of the same society and nation or are living in the same area [18].

The use of natural language is well firm in human culture. The number of natural languages that have emerged in the world is between 5000 to 7000, a large proportion of these languages have become extinct, and the extinction of languages will continue to happen by 2100 to approximately (50% to 90%), according to the estimates of the relevant studies in this area. The six most widely spoken languages in the world are English, Arabic, Chinese, Spanish, Russian and French where it approved within the United Nations [20].

After studying the concept of culture and analysing a set of elements that considered as a criteria related to culture factor which proposed as one of the Software Quality Factor (SQF), we found that one of these criteria is the natural language of the software system, which is a critical pillar in the development of the global software system, In this paper, we proposed that the natural language is on the cultural factors of the software systems.

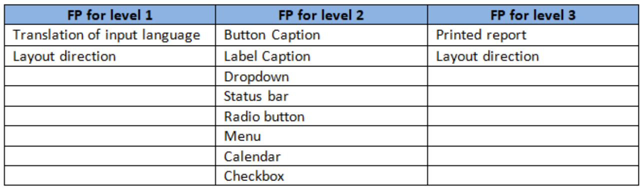
To measure and evaluate the software system regarding supporting the natural language as a cultural factor, we proposed a set of rules that must abide by to help evaluate the quality of the software system regarding the natural language as follows:

1. Setting several languages that the software system offers, and they divided into two parts:
   - The availability of the software system in the six international languages which are: Arabic, Chinese, English, Spanish, Russian and French.
   - The availability of the software system in other languages; in this case, we will give language less weight than the national natural language.

2. Setting the level of the natural language that is available in the software system. We classified this into three main categories to indicate the level of natural language in the software system as follows:
   - The first level: if the software system in this level allows the user to write and enter a linguistic text in the user interface using the language desired to be measured.
   - The second level: the availability of the user interface in the natural language entirely, which will also evaluate. In this case, the level of the natural language is higher than the first level.
   - The third level: If the system allowed the translation of all outputs to the natural language that will evaluate. In this case, the level of the natural language will be higher than the second level.

In this section, we used function point as a method for measuring the size and productivity of software systems based on outputs, inputs, queries, internal files and interfaces. Table 3 shows the check-point of the function for each level of the languages levels.

Table 3: Function Point for each level of natural languages



FP for level 1	FP for level 2	FP for level 3
Translation of input language	Button Caption	Printed report
Layout direction	Label Caption	Layout direction
	Dropdown	
	Status bar	
	Radio button	
	Menu	
	Calendar	
	Checkbox	

Information Flow for the Language (IF4L): IF4L Equation is an adaptation of a language in a system. The IF4L defined as; square the result of the Number of Input Language (NIL) by the Number of Output Language (NOL) in the system, as shown in Equations 1 and Equations 2.

$$IF4L = (NIL * NOL)^2 \quad (1)$$

$$WIF4L = (NIL * NOL * Weight)^2 \quad (2)$$

Where in Equation 2, if the language used was the United Nation Language, then the weight =2. In case of another language, it will =1.

For large systems that use several languages, a summation used to measure all languages in the system, as shown in Equations 3 and Equations 4:

$$IF4AL = \sum_{1=1}^{n} IF4L \quad (3)$$

$$WIF4AL = \sum_{1=1}^{n} WIF4L \quad (4)$$

Where n is the number of natural languages used in the system.

Information Flow for a Level of Language (IF4LL): IF4LL

Equations is an adaptation of a level of natural language in a system that consists of three levels, which are: input text level, user interface level and output level, and they calculated as follows:

The Level Of Input Text (LOIT) defined as square the result of the Number Of Input Text (NOIT) by its weight (W); as shown as follows in Equations 5.

$$LOIT = (NOIT * W)^2 \quad (5)$$

The Level Of User Interface (LOUI) defined as square the result of the Number Of User Interface (NOUI) by its weight (W); as shown in Equations 6.

$$LOUI = (NOUI * W)^2 \quad (6)$$

The Level Of Output (LOO) defined as square the result of the Number Of Output (NOO) and its weight (W); as shown in Equations 7.

$$LOO = (NOO * W)^2 \quad (7)$$

The weight (W) of each level $=2^n$Table 10 represents the corresponding n for each level.

Table 4: Value of N for Each Level

Levels	n
Level Of Input Text	1
Level Of User Interface	2
Level Of Output	3

Now, to calculate the IF4LL for any software system, the software system will have several levels from the previous levels for each natural language, so we have to use the following Equations:

$$IF4LL = LOIT + LOUI + LOO \quad (8)$$

For the large systems which use several languages, each has a specific level; a summation of the measures of all levels of languages contained in the system obtained as shown in the following Equations:

$$IF4ALL = \sum_{1=1}^{n} IF4LL \qquad (9)$$

Where n is the number of natural languages used in the system.

3.2 Religion Factor

Religion defined as a term used to talk about the set of ideas and doctrines that clarify the purpose of life and universe, also associated with the morals, ethics, theology, practices and actions related to certain beliefs [20].

Religion arranges the behaviour of people within a group of instructions and orders that people formulate, that is why everyone has a group of personal beliefs but religion one of the essential rights of individuals including the rituals whether symbols. Celebrations or festivals of the religion are respect for religion. Also, he respects all the aspects of cultural life organised by the religion followed by the group. There is a variation among groups by geographical areas regarding religion and beliefs.

Religion scholars have pointed out that religion is associated with institutions such as the family, hospitals, education and government, as it regulates different aspects of life [21].

With the software revolution and it is widespread within all public institutions, it became necessary to subject this software to the respect of users' beliefs and religions. Many researchers have discussed certain aspects of religion and its integration into technology.

Abokhodair and Vieweg presented a study on privacy as an essential element in the Islamic religion and its integration in social networking sites where he pointed out that religion and language are critical factors in the acceptance of technology and use in the Arabian Gulf, where privacy is a fundamental pillar in the Islamic religion. Privacy linked to a set of standards and rules emanating from Islamic religion, where the importance of privacy reflected in the preservation of the body, honour and reputation [22].

Also, Albugami and Ahmed discussed the challenge between the discovery of the cultural and religious boundaries that affect the Saudi Arabian approval for the information and communication technology. Especially that Saudi Arabia is considered a religiously conservative country [23].

Albugami and Ahmed have explained a set of concepts, including the improper use of the internet that does not comply with the Islamic values and norms of the Saudi Arabian society, and also, the need to keep up with the information and communication technologies to merge them with the educational process, which is considered a challenge for the government [23].

Through the literature review, it seemed that the software systems considered as a part of most institutions, in which that, the willingness or reluctance to use those systems is affected by the religions and beliefs of the users since they are part of their culture, in which users will be able to accept such systems that are adherent to their religions and beliefs.

Religion factors are a necessity to take into consideration to have a global software system that is not only limited to a particular group of users but also reflects the culture quality of the system.

In this chapter, based on analysing the cultural factors, religion proposed to be one of those factors. Therefore, we need a quantitative measurement to help evaluate and measure the quality of the software system, religion-wise.

We should adhere to a set of rules, to measure and evaluate religion factor, as following:

1. If the software system does not favour any specific religion by using any function point such as symbols, pictures, videos, advice or offences to any religion on the user interface, this means that the software system is suitable for all various users regarding their religion.
   - If the software system pointed out of specific religion or belief, then there are two cases:
2. The support of the system to the religion: in this case, we will measure to what extinct did the software system took consideration of the religion factor. The five most common religions in this study are Islam, Christianity, Buddhism, Hinduism, and Judaism. We categorised this support into two levels:
   - The first level: supporting a specific religion using any function point such as pictures, symbols or words that show on the user interface of the software system.
   - The second level: This level supporting a specific religion by using software functions. These functions include offering advice using texts or videos that display on the user interface of the software system. In this level, the measure of the software system will be more than the first level.
3. Offending a religion: we will measure to what extinct did the software system offend the religion. We categorised it into levels:

- The first level: Offending a specific religion using function point such as pictures, symbols or words on the user interface of the software system.
- The second level: Offending a specific religion by using function point such as videos or texts that show on the user interface of the software system. In this level, the measure of the software system will be more than the first level.

Information Flow for a Religion (IF4R): IF4R is an Equation that is an adaptation of a Religion in a system. The IF4R defined as square the result of the Number Of Function points that refer to Religion (NOFR) by its weight(W) which represented in table 12, as shown in Equations 10.

$$IF4R = (NOFR * W)^2 \quad (10)$$

The weight $W=2^n$ where the value of n as shown in Table 4.
We used a summation of measurement all the function points of the religion which was in the system, as shown in Equations 11.

$$IF4AR = \sum_{1=1}^{n} NOFR \quad (11)$$

Where n is the number of religion function points in the system.

Table 5: Value n for Each Level

FP for each Level	n
Symbol, Word, Picture	1
Video, Text	2

3.3 Social Habits, Customs, Ethics, Customs and Law Factor

Henrich defined Social habits as practiced among people living in the same geographical areas when they interact together [24]. Manly defined customs as a traditional and widely accepted way of behaving or doing something specific to a particular society, place, or time [11]. Driskill defined ethics as moral principles that govern the behaviours of people either in groups or individuals [26]. Nelken defined law as the systems and rules offered by governments of countries to control and govern its peoples, citizens and members behaviours [7].

Aversano and Guardabascio studied causes of success Complier's the open source ERP system. They found that it handles the administrative matters of commercial enterprises globally, there are 39 languages available, and it is compatible users all over the world with different their culture. The user can use the system with few geographical barriers. Various language versions can also be obtained from the web store and customised to suit any language requirement for different tax laws, accounting rules and currencies, thus spreading globally, which users can use anywhere in the world without paying attention to geographical distances and obstacles. Thus increasing the system's success and global, this increases the trust of the user in the system reach [28].
Tencent Company had taken advantage of the social habits in china and many different East Asian and Southeast Asian countries to release a digital version of a social habit that widely spread there, and it is called the red bag, it released within WeChat application. In this caused increase acceptance users of the WeChat application that reached two million users in fifteen days [8].
On the other hand, the Chinese government shut down many social media applications like Twitter, to keep the cultural and social values and legacies from fading away by the invasion of other cultures [8].

By the literature review and by analysing the effect of the previous four culture factors on the success and failure of many software systems, we considered each of the social habits, customs, ethics and law factors as a cultural-quality criterion of the software system.
To measure and evaluate the quality of software system regarding the social habits, customs, ethics and law of the user prevalent in a particular culture, we suggested a group of rules as follows:

1. The software system disregarded the social habits, customs, ethics and law of the user, in this case, the measure of the software systems decrease.
2. To determine if the software system supported and took consideration of any of the social habits, customs, ethics and law of the user, and there are two levels for this case:
   - The first level: The software system included function points such as pictures, symbols or words that support any of the social habits, customs, ethics and law, without applying or implementing them directly and actually in the system.
   - The second level: The system directly supports any of the social habits, customs, ethics or law by applying or implementing them in the software systems. In this case, the measure of the software system will be more than the first level.
3. The determination whether the software system offends any of the social habits, customs, ethics and law of the user, in this case, we have two levels, as follows :

The first level: If the system directly offended any of the social habits, customs, ethics and law of the user by using pictures, symbols, video, or words within the software system.
The second level: If the system directly offended any of the social habits, customs, ethics or law by applying or implementation them in the software systems. In this case, the measure of the software system will be more than the first level.

Information Flow for a Culture x (IF4Cx): x represents the factors culture which, are social habits, customs, ethics and law. IF4Cx Equations is an adaptation of previous culture factors in a system. The IF4Cx defined as, square the result of the Number Of Function points of the Culture x (NOFCx) multiply by weight plus (NOACx) multiply by its weights (W).
If the software system has supported any of these factors, we apply Equations 12.Table 5 represents weight (W) for each level.

$$IF4Cx = (NOFCx * W)^2 + (NOACx * W)^2 \quad (12)$$

The weight $W =2^n$, where value of n is shown in Table 5.
If the software system has offended any of these factors, this means that the result of the applied Equations will be negative, as shown in Equations13.

$$IF4Cx = -(NOFCx * W)^2 + (NOACx * W)^2 \quad (13)$$

Table 5: Value of N for Each Level of culture x

Levels	n
Function points that pointed to any of the Culture x factors	1
Culture x factors that were applied in the SW	2

4. RESOLT AND DISCUSSION

We evaluate the proposed culture factors on three commercial software systems, namely: Smart Soft Pharmacy application, MTA application and School Root application.

Table 6 shows the results of applying the equations of culture on the three applications, which are: Smart Soft Pharmacy, MTA Application, and School Root Systems.

Table 6: Equations Results of Case Studies

Application Name	Natural Language Metrics		Religion Metrics	Culture x Metrics
	WIF4AL	IF4ALL	IF4R	IF4Cx
Smart Soft Pharmacy	500	33312	0	20
MTA application	200	1440	84	0
School Root System	200	5392	0	20

Figure 2 shows the equations results for Smart Pharmacy Application. According to the proposed equation, the Smart Pharmacy Application achieved a higher quality regarding Language, Religion and Law factors.

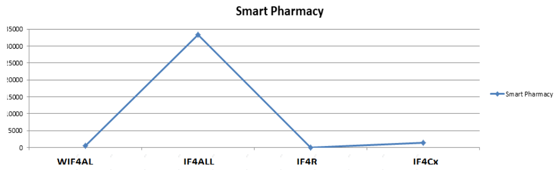


Figure 2: Equations Results for Smart Pharmacy Application

Figure 3 shows the Equations results for MTA Application. The quality of MTA was high regarding language and law, but it was low regarding religion factor.

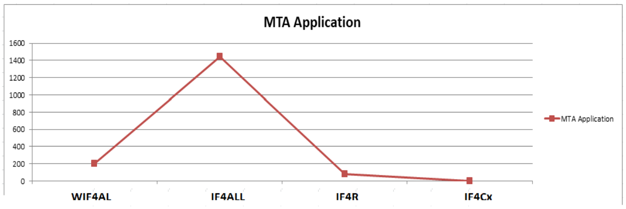


Figure 3: Equations Results for MTA Application

Figure 4 shows the Equations results for School Root System. The qualities of the School Root System are highly regarding law, religion and language factors.

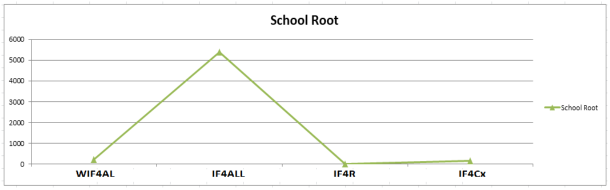


Figure 4: Equations Results for School Root System

Figure 5 shows all the results for the case studies proposed in this paper. It showed that the Smart Soft Pharmacy application achieved the highest degree of the quality regarding the natural language factor in comparison with other School Root System and MTA application, in which the higher the value of the scale that we got from the Equations, the higher quality of the system regarding the natural language factor.

MTA application achieved the least degree of the quality regarding the religion factor in comparison with the other School Root System and Smart Soft Pharmacy application, in which the higher value of the scale that we got from the religion Equations, which decreases the quality of the software systems regarding religion factor, therefore leads to its failure.

Furthermore, finally, the quality of the School Root System and Smart Soft Pharmacy application has risen regarding the law factor, where the value of these scales, when applying the law Equations of these systems, was higher than its value in the MTA application which equal to zero, and it did not take into consideration any of the social habits, customs, ethical and law factors.

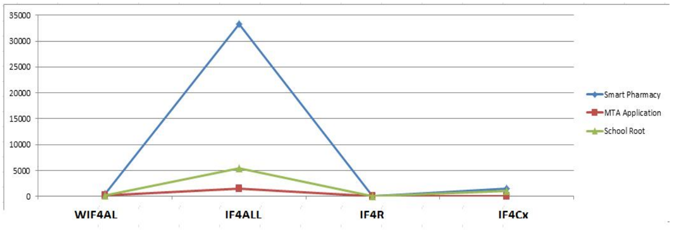


Figure 5: Equations Results for Case Studies

5. CONCLUSION

In this paper, we study the failure and success software and emerging the software quality models to reduce the failure software. Hence, we discussed the software quality models for the presence of cultural and social requirements. This paper compares the quality model factors from cultural and social aspects. Furthermore goes behind the definitions of the cultural requirements form the software quality factors, sub-factors and criteria that affect the software failure and success.

Furthermore, new factors proposed to get clear and accurate differences between software quality models. This method requires to assigning values for the sub-factors moreover the main factors, which is giving a clear picture of the differences between the models.

The values in this study were given equivalently between the factors and between the sub-factors. In a specific domain, the cost for each factor and sub-factor has to define according to the selected domain. Eight cultural criterion factors proposed for satisfactions of software failure and permanence variables. To measuring the complexity of cultural factor we proposed cultural Equations. Finally a case study is represented in a result and discussion.

REFERENCES

1. Youness, B., Abdelaziz, M., Habib, B., & Hicham, M. (2013). "**Comparative Study of Software Quality Models**". IJCSI International Journal of Computer Science Issues, 10(6), 1694-0814.
2. Ward Jr, W. A., & Venkataraman, B. (1999). "**Some observations on software quality**", Proceedings of the 37th annual Southeast regional conference. https://doi.org/10.1145/306363.306367
3. Al-Badareen, A. B., Selamat, M. H., Jabar, M. A., Din, J., & Turaev, S. (2011). "**Software quality models: A**

comparative study". International Conference on Software Engineering and Computer Systems. https://doi.org/10.1007/978-3-642-22170-5_4

4. Tabassum, A., Bhatti, D. S. N., Asghar, A. R., Manzoor, I., & Alam, I. (2017). "**Optimized quality model for agile development: extreme programming (XP) as a case scenario**". IJACSA: International Journal of Advanced Computer Science and Applications, 8(4), 392-400. https://doi.org/10.14569/IJACSA.2017.080453
5. Agarwal, N& ,.Rathod, U. (2006). "**Defining 'success' for software projects: An exploratory revelation**". International journal of project management, 24(4), 358-370.
6. Spencer-Oatey, H., & Franklin, P. (2012). "**What is culture?**", A compilation of quotations GlobalPAD Core Concepts, 1-22.
7. Aversano, L., Guardabascio, D., & Tortorella, M. (2017). "**Analysis of the documentation of ERP software projects**". Procedia computer science, 121, 423-430. https://doi.org/10.1016/j.procs.2017.11.057
8. Holmes, K., et al. (2015). "**Red Bags and WeChat (Wēixìn): Online collectivism during massive Chinese cultural events**." Global Media Journal: Australian Edition9(1): 15-26.
9. Mukhra, R. et al. (2017). "'**Blue Whale Challenge': A game or crime?**" Science and engineering ethics: 1-7.
10. Hamed Fawareh "**Detection a Design Pattern through Merge Static and Dynamic Analysis Using Altova and Lambdes Tools**" International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 8518-8522.
11. Manly, S. (2017). "**Language, custom and nation in the 1790s**". Locke, Tooke, Wordsworth, Edgeworth: Routledge. https://doi.org/10.4324/9781351154284
12. McCall, J. A., et al. (1977). "**Factors in software quality**". Volume i. concepts and definitions of software quality, GENERAL ELECTRIC CO SUNNYVALE CA.
13. Boehm, B. W., et al. (1978). "**Characteristics of software quality.**"
14. Dromey, R. G. (1995). "**A model for software product quality.**" IEEE Transactions on software engineering 21(2): 146-162.
15. Wolski, M., et al. (2018). "**Software quality model for a research□driven organisation—An experience report**." Journal of Software: Evolution and Process 30(5): e1911.
16. Esaki, K., et al. (2012). "**Introduction of quality requirement and evaluation based on ISO/IEC square series of standard**". International Conference on Trustworthy Computing and Services, Springer.
17. O'Reilly III, C. A., Caldwell, D. F., Chatman, J. A., & Doerr, B. (2014). "**The promise and problems of organizational culture: CEO personality, culture, and firm performance"**. Group & Organization Management, 39(6), 595-625. https://doi.org/10.1177/1059601114550713
18. Santana, C .(2016) ."**What Is Language?**" Ergo, an Open Access Journal of Philosophy, Vol. 3.
19. United nation web page: https://www.un.org/en/. Last visit 13/1/2020.
20. Forbes, B. D., & Mahan, J. H. (2017). "**Religion and popular culture in America**" Univ of California Press.
21. Monaghan, J., & Just, P. (2000). "**Social and cultural anthropology: A very short introduction**". Vol. 15, Oxford Paperbacks.
22. Abokhodair, N., & Vieweg, S. (2016). "**Privacy & social media in the context of the Arab Gulf**". Proceedings of the 2016 ACM Conference on Designing Interactive Systems. https://doi.org/10.1145/2901790.2901873
23. Albugami, S. S., & Ahmed, V. (2016). "**Effects of culture and religion on the use of ICT in the Saudi education system**". IIER: International Institute of Engineers and Researchers. 25-27.
24. Henrich, J. (2015). "**Culture and social behavior**". Current Opinion in Behavioral Sciences, 3, 84-89
25. Manly, S. (2017). "**Language, custom and nation in the 1790s**". Locke, Tooke, Wordsworth, Edgeworth: Routledge. https://doi.org/10.4324/9780429427473
26. Driskill, G. W. (2018). "**Organizational culture in action: A cultural analysis**", workbook: Routledge.
27. Nelken, D. (2017). "**Beyond law in context: developing a sociological understanding of law**", Routledge. https://doi.org/10.4324/9781315261621