Pakistan Journal of Engineering and Technology
ISSN (Print): 2664-2042; ISSN (Online): 2664-2050
Received: 2024-01-09
Accepted: 2025-03-06
https://doi.org/10.51846/vol8iss1pp55-63

PAKJET
Pakistan Journal of Engineering & Technology

Article

Effects of Layered Architecture & UI Design upon Maintainability & Usability: An Implementation Case Study of 3D Electronic Communication Surveillance Analytic

Shazia Yousaf [1],*, Hina Ibrar [1], Sobia Yousaf [1], and Tauseef Ahmad Rana [2]

[1] Department of Software Engineering, National University of Modern Languages (NUML), Rawalpindi, Pakistan
[2] Department of Computer Software Engineering, National University of Sciences and Technology, Islamabad, Pakistan
* Correspondence: Shazia Yousaf (shazia.yousaf@numl.edu.pk)

Abstract: Software Architecture provides views and structure of the system at a high-level of abstraction. The selection of software architecture and user interface design has a great influence upon the quality of software. Our research is going in two directions; one to find the impact of layered architectural style upon maintainability of software and secondly to find the impact of user interface design with three-dimensional (3D) modeling upon usability of software. We need to choose architectural solutions based on the proven facts in order to find the desired quality attributes. The literature for the impact of architecture patterns on quality attributes shows a lack in achieving usability, that's why we have chosen to assess a software architectural pattern along with interface design in order to achieve usability along with other quality attributes. We implemented a system "Electronic Communication Surveillance Analytic tool with 3D Modeling" based on the layered architectural style, then we evaluated this system for maintainability and usability.

Keywords: 3D Modeling, Layered Architecture, Maintainability, Quality Attribute, Software Architectural Pattern, User Interface Design, Usability

1. Introduction

1.1. Background and Motivation

Software Architecture provides views and structure of the system at high-level of abstraction. The selection of architectural style has direct impact on quality attributes of software. Fulfilment of non-functional requirements also based on architectural pattern chosen [1]. It's determined by the system architecture whether the system will fulfil the main requirements or not [2]. Architectural pattern chosen decides the provision of quality areas or no functional requirements of the system through the major components and connectors. Furthermore, these patterns also target specific set of quality areas attributes. Hence, designing a sound architecture of the system gives us two-fold benefits; one clear understanding of the system by providing the structure of the system, and secondly takes the responsibility for quality requirements to be satisfied. For a successful software the requirements of the system must be satisfied. These requirements include both functional and quality requirements. Thus we can test the fulfilment of these quality goals incorporated by an architecture style chosen as quality requirements satisfaction of the system being developed [3]. In other words, choice of architectural solutions relays on satisfying the required set of quality [4]. Selecting the right architectural solutions verify to be vibrant to have a vast impact on the overall success of the system. The key goal of this research is to discover the impact of selecting layered architectural pattern and user interface design on the quality attributes, specifically usability and maintainability.

Meeting the user needs and getting the user satisfied is one of the major goal of getting a good quality software. In other words, the quality of software is also essential for satisfying the customer expectations and demands. The quality of a software is measured in terms of usability, efficiency, learnability and many more. Among these, usability is the main quality attribute of a software. Usability comes along with appropriateness, recognisability, learnability, operability, user error protection, user interface aesthetics, and accessibility [5].

High maintainability can reduce almost 75% of the cost of utmost systems' life cycle. Software maintainability is well-defined as the how easy is it to modify a system in order to correct faults, increase performance or other attributes, or familiarize to a different environment [6]. Maintainability is defined in terms of modularity, reusability, analyzability, modifiability, and testability [5].

The literature [3], [7]– [10] for impact of architecture patterns on quality attributes shows lack in achieving usability that's why we have chosen to assess a software architectural pattern along with interface design in order to achieve usability along with other quality attributes like security, maintainability, portability, performance and reliability which these architectural styles are already providing. The aim of this research is to find impact of layered architecture along with user interface design upon required quality attributes. Therefore, we implemented a system, “Electronic Communication Surveillance Analytic tool with 3D Modeling” to achieve a required set of quality attributes based on the selected architectural style. Through an implementation case study, the usefulness of the proposed approach is established in attaining maintainability and usability via plotting relationship impact among layered architecture, user interface design and these two quality attributes.

Intelligence and defense agencies need to quickly identify a criminal who is involved in smuggling, terrorism, thefts and robberies, etc., for which they access electronic communication records of various people. They try to find the suspected person based on this information. Analyzing all these records can become difficult task when dealing with tens or hundreds of persons’ emails, messages or call records. Creating a 3D interactive application that displays all the persons along with the people they are in contact with would help quickly discovering contact patterns among various persons of interest. The 3D representation gives the system a real world look and feel.

As per our knowledge, such surveillance systems in use in Pakistan have two dimensional (2D) interface. Some related existing bulk information analysis tools include the followings

- **Liveplasma:**

It is basically a movies search engine where you can search by the name of movie, band, and artist or by the name of director [11]. The interface of this website is built on 3D modeling which provides you visualization of movies and music.

- **Arnetminer:**

It is a searching website built using the 3D modeling which provides you the researcher’s profile along with the publications. In fact, it is a freely available online service used for indexing and searching educational social networks [12].

The main focus of this research is to check how layered architecture affects the software maintainability and how the application of 3D modeling in data presentation impacts the software usability. The research studies the impact of layered architecture on key maintainability attributes, such as modularity, testability, reusability, modifiability, and analyzability, by checking how such an approach improves the software organization, testing efficiency, component reusability, ease of modification, and system analysis. The study also aims to determine the impact of 3D modeling on various aspects of usability, including appropriateness, recognizability, learnability, operability, user error protection, user interface aesthetics, and accessibility. It specifically analyzes how 3D models facilitate the visualization of complicated data, help in discovering patterns in data, make user interaction easier and enhance aesthetics. The study hence combines these two aspects into one to provide an inclusive understanding of how layered architecture and 3D modeling together contribute to the usability and maintainability of a software system.

1.2. Literature Review

Overall quality of the system is measured by the fulfilment of non-functional requirements of a software system. Research in the area of software quality has stated many quality models collected by a set of quality attributes [13]. ISO/IEC 25010 includes a set of eight primary quality attributes (Functionality, Security, Reliability, Compatibility, Usability, Performance-Efficiency, Maintainability, and Portability), each with a secondary set of quality attributes [14].

Software design and architecture have the great impact on the success of software. Richard N. Taylor and Andre Van stated: Designing of software is an activity that cannot be successfully off-shored and design can determine an organization’s success [15].

Architectural patterns are refillable solutions to the reoccurring architectural problems. There is a list of many architecture patterns which are proposed over time for designing software architecture [3], [4], [7], [16]. Few reside in the classification of domain specific architecture patterns while few are independent [7], [8]. Due to space limitations, in this paper, we have used only a limited number of architecture pattern.

Yair Viveros et al. have proposed a Layered Software Architecture for developing video surveillance systems [17]. L. A. Rivera Alvarado shows another interesting use of the same Software Architecture style in the software development of Moveable Learning Objects using augmented reality [18].

Quality attributes are directly affected by the chosen software architectural pattern shown in Table 1. Studies to find the relational impact of quality attributes and architecture patterns and are established in [3], [8], [19]. In order to find the influence of architecture patterns on target software system we can check the impact of the architectural style on the quality attributes.

Table 1. Mapping table of the relation of architecture patterns with quality attributes showing the impact factors [3], [7]–[10].

Architecture Patterns	Quality Attributes					
	Reliability	Efficiency-performance	Usability	Security	Maintainability	Portability
Layers	+1	-1	0	+1	+1	+1
Client Server	-1	-1	0	+1	-1	-1
Pipe and Filters	-1	+1	-1	-1	+1	-1
Blackboard	-1	-1	0	-1	+1	-1
Microkernel	+1	-1	0	0	+1	0
Broker	0	-1	0	-1	+1	+1
MVC	0	-1	+1	0	+1	+1

Clashes on correlation impact of architecture patterns on quality attributes were identified from the literature review. So, the relative impact might not be 100% accurate because of differences in diverse studies. But in order to be specific to our approach we are focusing on the above-provided facts from literature. Here we can clearly see that architectural patterns are lacking the usability. Therefore, we need to look for some user interface design along with an architectural style in order to achieve the quality attributes, including usability.

1.2.1. Comparative Analysis with Existing Techniques

In the literature, there are numerous studies that have explored the relationship between quality attributes and architectural styles. Harrison et al [3] used the qualitative approach without implementation to map the layered architecture with performance and modifiability attributes.

Fielding [7] particularly focus on highlighting the scalability and simplicity attributes using the web-based architectural styles. Moreover, they don't consider the UI and usability. Majidi et al [8] and Eftekhari et al [9] emphasised on reusability, maintainability and performance by exploring the classification framework and matrix evaluation of architectural styles. They relied only on static analysis and didn't use empirical validation.

Buschmann and Henney [10] introduced the MVC and layered architectural styles. They are also called the reusable architectural styles. They focused on maintainability and modularity without using the role of UI design. Comparison of these studies is shown in Table 2.

Table 2. Comparison between this study and studies [3], [7]– [10].

Criteria	This Study (3D-ECSA)	[3]	[7]	[8]	[9]	[10]
Architecture Type	Layered	Layered, Pipes & Filters, Broker	REST (Representational State Transfer)	Client-server, Blackboard	Layered, Repository	Layered, MVC, Broker
Evaluation Method	Case Study (Implemented System)	Qualitative pattern-attribute mapping	Formal modeling	Literature Survey	Matrix-based evaluation	Pattern catalog with examples
Quality Attributes Focused	Usability, Maintainability	Modifiability, Performance, Availability	Scalability, Modifiability	Broad QA range	Reusability, Performance, Security	Pattern consequences (implicit QAs)
UI Consideration	Yes	No	No	No	No	Minimal
Validation	Real-world Tool (3D-ECSA)	No	No	No	No	Partial (pattern examples)

By implementing the 3D-ECSA tool, our work not only exhibits practical applicability but also represents impact relationships between architectural style and quality attributes. The present study has integrated layered architectural style with UI design to target the usability and maintainability as quality attributes.

1.3. Contribution

This study presents a novel approach for improved quality by making the following contributions:

- It integrates layered architectural design with user interface (UI) design to achieve the quality attributes of usability and maintainability.
- It checks how layered architecture affects the software maintainability
- The development of a standalone desktop application, the 3D Electronic Communication Surveillance Analytic (3D-ECSA) tool, demonstrating its practical applicability.
- It specifically analyzes how 3D models facilitate the visualization of complicated data, help in discovering patterns in data, make user interaction easier and enhance aesthetics.
- The study hence combines these two aspects into one to provide an inclusive understanding of how layered architecture and 3D modeling together contribute to the usability and maintainability of a software system.

2. Materials and Methods

Our approach uses a layered architectural design pattern along with user interface design to achieve the required set of quality attributes through mapping the impact relationship among layered architecture, UI design and quality attributes. Fig. 1 shows the system design of our proposed approach.

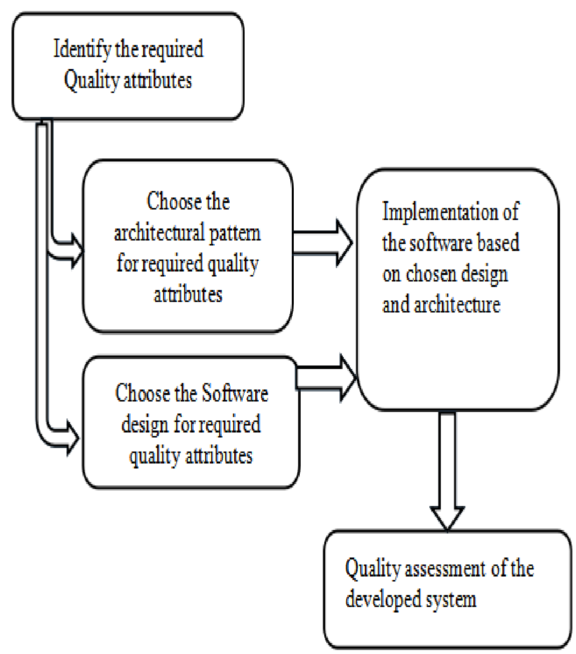

Figure 1. Proposed 3D-ECSA.

We have developed an Electronic Communication Surveillance Analytic tool with a 3D Interface to provide quick and efficient analysis and review of electronic communications (via phone calls, emails, and text messages) among different people in a three-dimensional (3D) space. Table 3 shows the features of our implemented tool.

Table 3. Features of electronic communication surveillance analytics.

Sr. #	Feature Name
1	Display of contacts and connections in a 3D environment
2	Generating Time analysis report
3	Generating a Date analysis report
4	Contacts Record Management
5	Communication History Management
6	View connection depth
7	Glow suspected connection
8	Filter connection
9	Import communication Records
10	Export communication Records

We have developed a standalone desktop tool which is an Electronic Communication Surveillance Analytic tool. We have used the following tools and technologies in the implementation of our system:

- Blender: an open source 3D modeling software
- Panda 3D: is actually a game designing engine and a tool for designing of 3D environments
- MySQL: is an open-source relational database management system
- SQL Alchemy: is the Python SQL toolkit
- Object Relational Mapper: provides an object-oriented layer between relational databases and object-oriented programming languages
- Python: is a high-level, general-purpose programming language.
- Tkinter widows designing library is also used in interfacing
- Google maps API: an application programming interface to retrieve data from Google Maps.

Fig. 2 shows class diagram of our system which may help you understand the major modules and their relationship among each other.

We built the layered architecture of 3D Electronic Communication Surveillance Analytic tool (3D-ECSA) (shown in Fig. 3) by dividing it into three kinds on components which are Graphical User Interface (GUI) for front end interface representation, Commands which are the providing the implementation of different modules of the system and Database layer which contains all the database files. Here we can see GUI is coordinating to Commands layer which is further linked to Database layer.

3. Results

We have used this system as a case study to demonstrate our proposed approach in order to find the best possible way for using Layered architecture along with User Interface (UI) design in order to achieve a specific set of quality attributes.

The main GUI of 3D-ECSA contains the menu bar with the multiple categories of options to view, analyze and manage the communication records of persons.

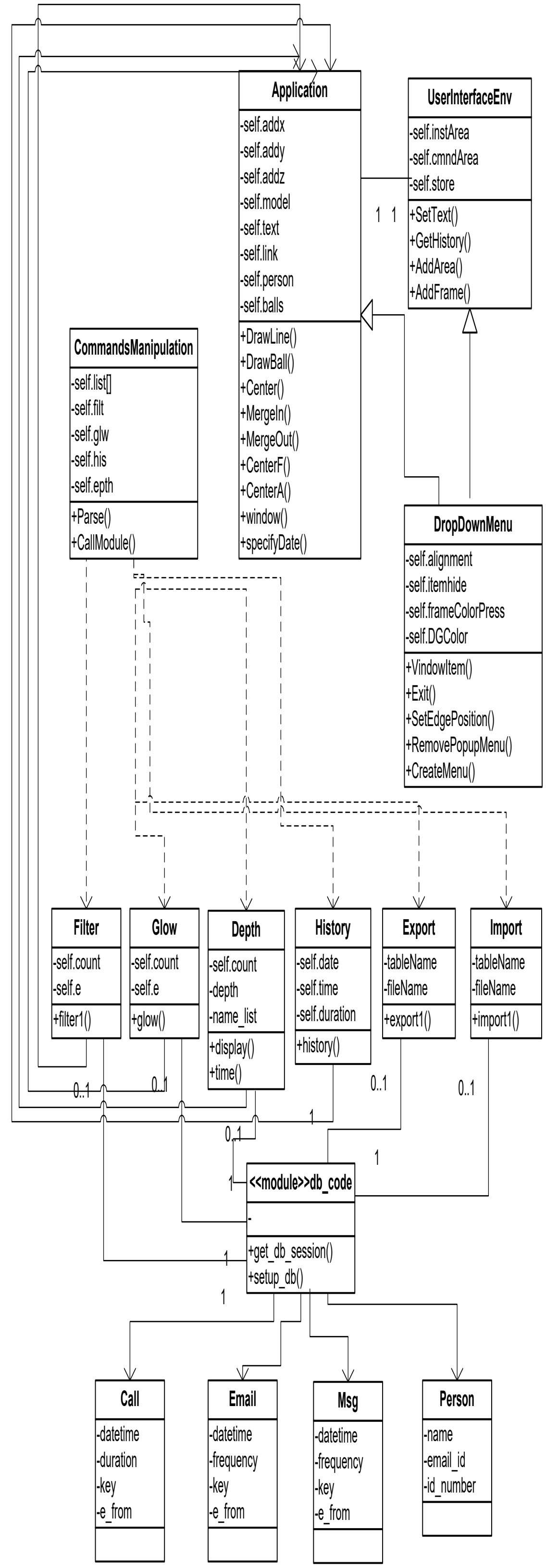

Figure 2. Class diagram of 3D ECSA.

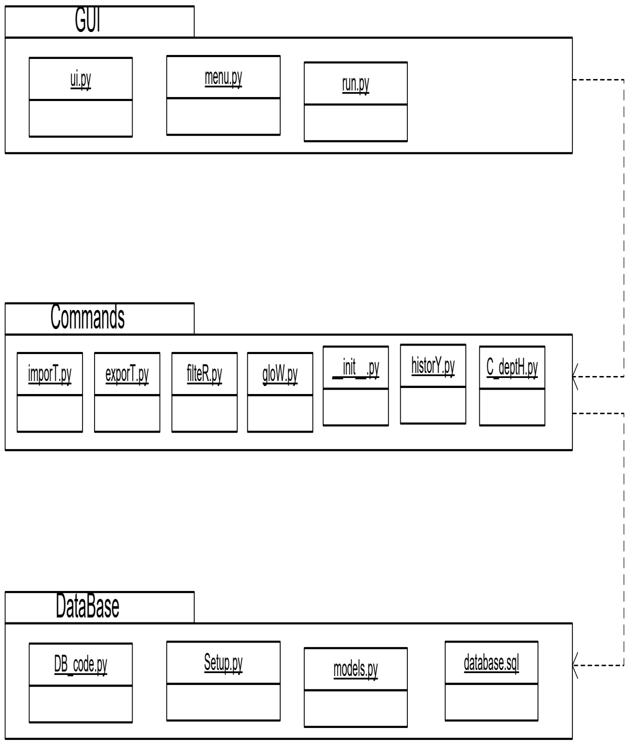

Figure 3. Layered architecture of 3D ECSA.

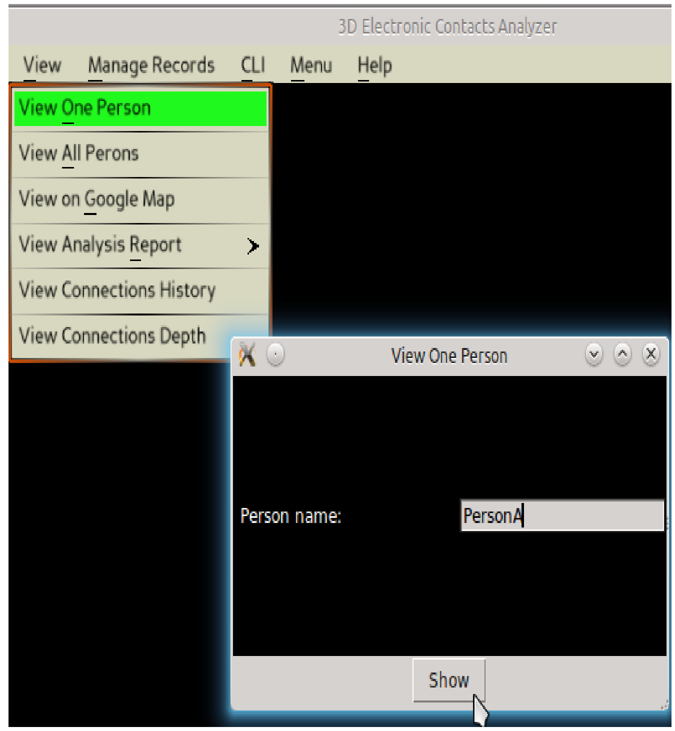

Figure 4. View a person from GUI.

Fig. 4 shows the screen which appears when user clicks the required option to View Contacts and Connections in order to perform further operations on them. Fig. 5 shows the result to view the person and his/her connection depth. It actually shows a person and his connections up-to required lower of depth in order to view who is connected with whom and that is further connected with whom.

We used dummy database for records of emails, messages and phone calls of imaginary persons. But in future the real time use may need the actual database records in real-time for this importing and exporting the communication records is the attractive and required feature.

Fig. 6 is showing the result GUI when the user wants to view the time analysis report. 3D-ECSA shows all records and connections along with distinguished display for suspected or alarming time.

The following image (see Fig. 7) shows the screenshot when user wants to import records (containing values of all the fields present in the specified database table) of data from external CSV file into database table.

3.1.1. Validation for Maintainability

As maintainability lets you add new modules into system without creating any changes in the existing modules. For prove of concepts we added two more modules in the system. Fig. 8 One is command line interface in order to manipulate all the features of the system through defined commands. Fig. 9 is the module to show the location of the person on Google Maps.

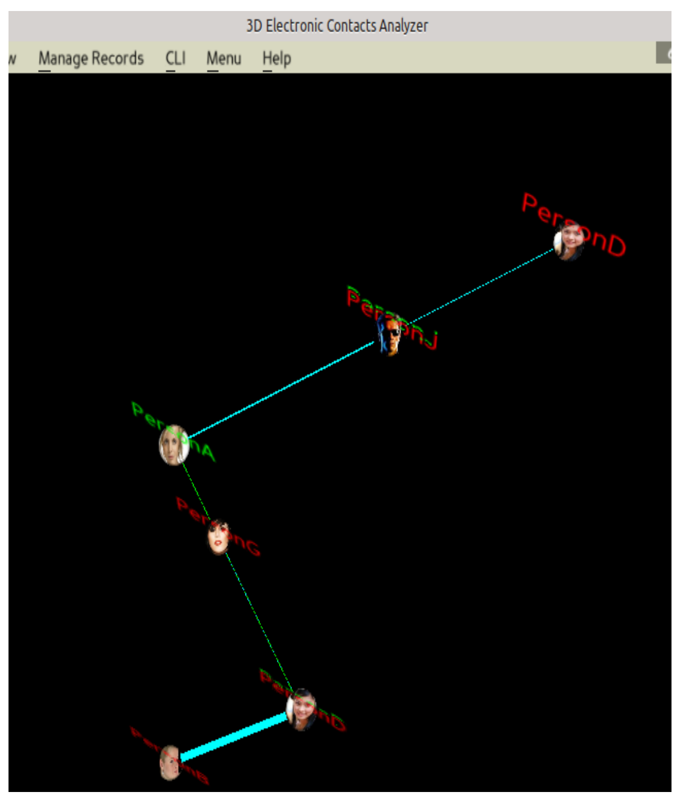

Figure 5. Result of a person record from GUI.

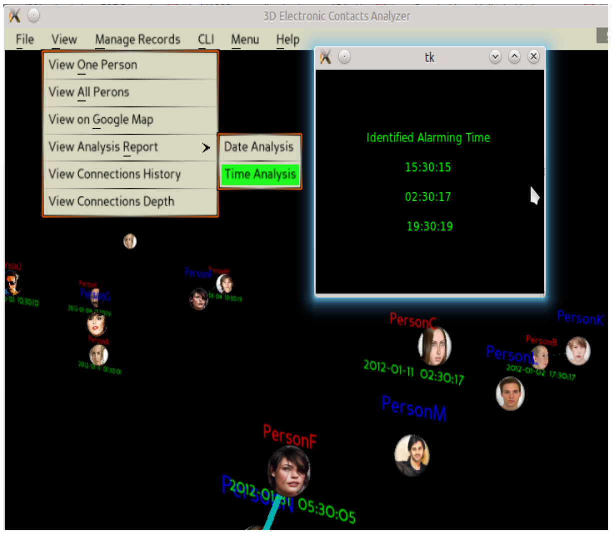

Figure 6. View time analysis report from GUI.

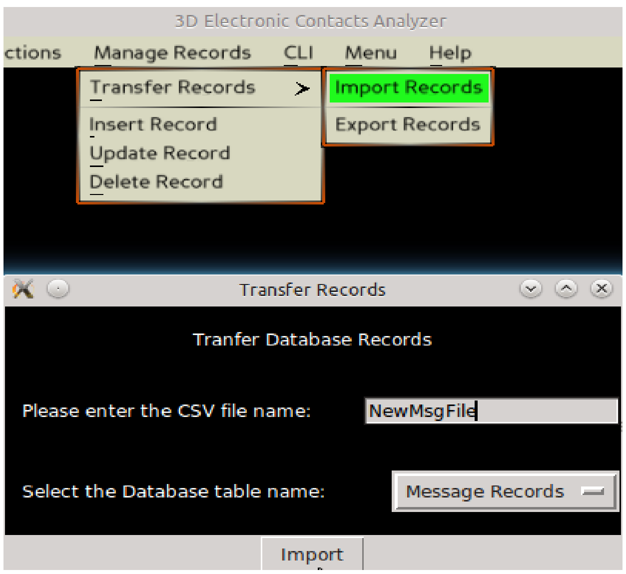

Figure 7. Import database records.

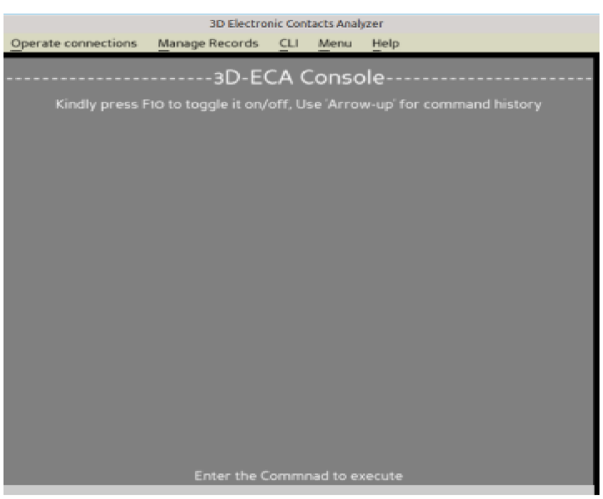

Figure 8. CLI of 3D-ECSA.

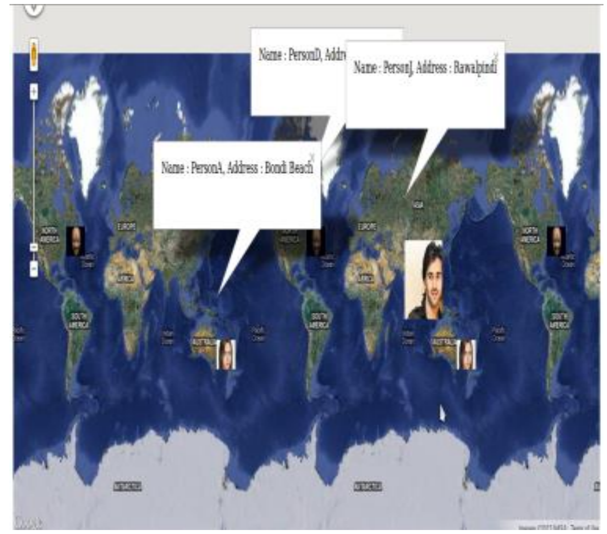

Figure 9. View contact and connection on Google Maps.

4. Quality Assessment Results

We got our system assessed for scalability and usability manually by us and by 5 professionals of the software engineering community shown in Table 4. They are basically academicians and researchers having more than five years of experience. The evaluators used the system for 5 days and were asked to rate the systems against different quality attributes whether the quality attributes exist in the system or not if exists rate it in percentage form. We also provided them the internal design and coding artefacts to evaluate the maintainability of 3D-ECSA. The quality attributes tested include the maintainability and usability.

Table 4. Quality attributes tested in 3D ECSA tool.

Maintainability	Usability
Modularity	Appropriateness
Reusability	Recognisability
Analyzability	Learnability
Modifiability	Operability
Testability	User Error Protection

4.1. Maintainability

The impact of layered architecture on the maintainability of software was evaluated by evaluating key maintainability attributes acknowledged by users. The graph (Fig. 10) displays the results of this evaluation, summarizing the users' responses on the effectiveness of layered architecture in enhancing various maintainability characteristics. The maintainability attributes evaluated include modularity, testability, reusability, modifiability, and analyzability. Fig. 10 shows the results of maintainability attributes recognized by the users in 3D-ECSA.

Fig. 10 shows the percentage rating against the quality attributes assessed. Users rated these attributes based on their experiences with software systems designed using a layered architecture approach. Table 5 shows the results of maintainability attributes and the metric values recognized by the users in 3D-ECSA.

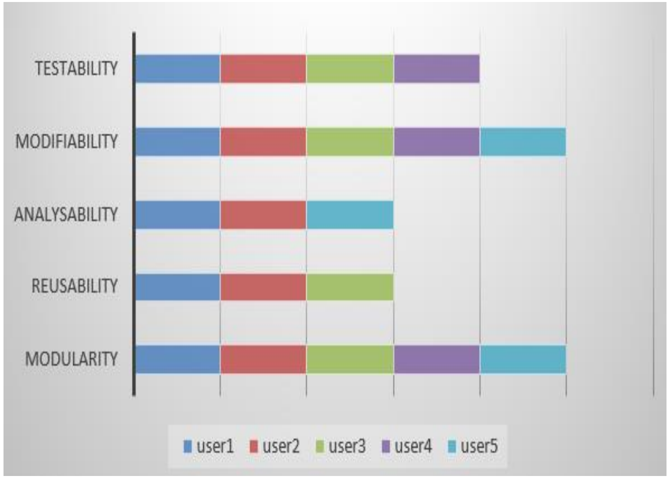

Figure 10. Maintainability assessment result.

Figure 11 is the graphical representation of the above-mentioned metrics values against the maintainability attributes recognized by the users in 3D-ECSA.

Modularity of software means the level to which parts of the system can be separated and then reassembled and was rated high by users. In addition, on the aspect of reusability, the layered architecture also proved quite positive. Users noticed the fact that components within single layers could be reused from one point to another in an application. Another attribute in which layered architecture showed considerable benefits was the analyzability of the software, or the ease with which the software can be analyzed for potential defects or performance bottlenecks. In terms of modifiability, layered architecture generally improved the ability to make changes to the system, though with some limitations. Users reported that changes could be implemented more efficiently when confined to specific layers without affecting other parts of the system. Modularity and modifiability got highest ranking among others. The graph (Fig. 10) reflects a high user rating for modularity and modifiability in systems employing layered architecture.

Table 5. Rating results for metrics of maintainability QA.

Quality Attributes	Modularity	Reusability	Analyzability	Modifiability	Testability
Metrics	**Coupling**	**Cohesion**	**Lines of code reused**	**Number of components reused**	**Code readability score**
Users					
User1	1	5	5	5	5
User2	1	5	5	4	5
User3	1	5	4	5	2
User4	1	5	2	3	2
User5	1	5	2	1	5

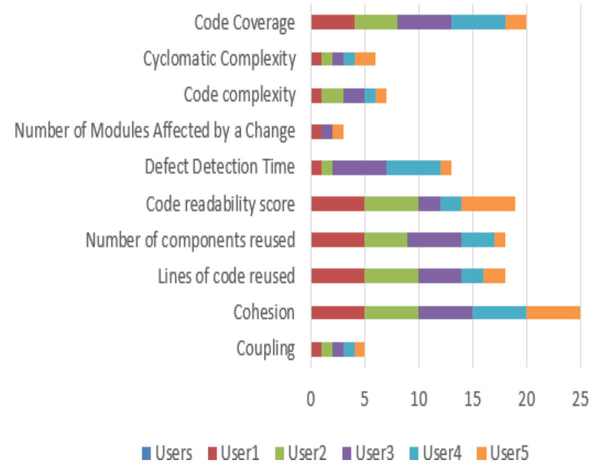

Figure 11. Maintainability assessment result.

4.2. Usability

This section sums up the results from the evaluation of 3D modeling's influence on the usability of software in relation to seven major attributes: Appropriateness, Recognizability, Learnability, Operability, User Error Protection, User Interface Aesthetics, and Accessibility.

The assessment of 3D modeling in data representation showed significant improvements in the usability of software on various key attributes. Appropriateness was improved since 3D models provided the user with a better way to visualize complex data which was lacking in data representation in tabular form. Recognizability was improved due to spatial depth and clear differentiation between persons, trends, and outliers in communication in terms of email, phone call and messages. Learnability was initially a little difficult because of the learning curve, but the users soon adapted to the interactive nature of the 3D models. Interactivity enhanced operability but sometimes made the model overly complex, which hindered ease of use. User Error Protection was enhanced by visual feedback and features like undo/redo, thus reducing the number of errors. User Interface Aesthetics were greatly improved to make the interface more engaging and visually appealing. Fig. 12 shows the results of usability attributes recognized by the users in 3D-ECSA. Fig. 12 is showing the percentage rating against the usability quality attributes assessed.

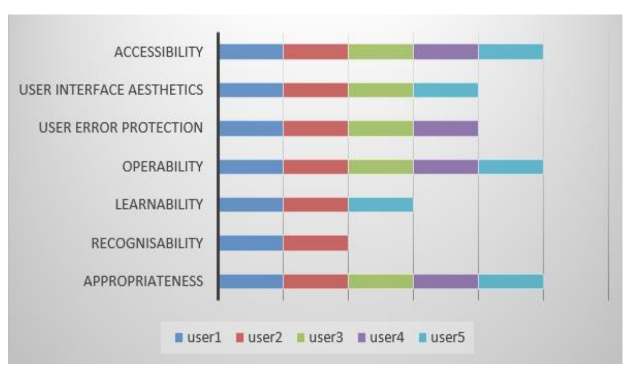

Figure 12. Usability assessment result.

Users rated these attributes based on their experiences with software systems designed using a layered architecture approach. Table 6 shows the results of usability attributes and the metrics values recognized by the users in 3D-ECSA.

Table 6. Rating results for metrics of usability QA.

Quality Attributes	Appropriateness	Recognisability	Learnability	Operability	User error protection	User interface aesthetics	Accessibility
Metrics / **Users**	**User Satisfaction**	**Contextual Fit**	**Search Success Rate**	**Training Time**	**Satisfaction with Initial Use**	**User Control and Freedom**	**Task Success Rate**
user1	5	5	5	3	3	5	4
user2	5	4	5	3	3	5	5
user3	5	4	3	5	5	5	2
user4	5	5	2	4	5	5	4
user5	5	5	1	3	2	5	5

Fig. 13 shows the graphical representation of the above-mentioned metrics values against the usability attributes recognized by the users in 3D-ECSA.

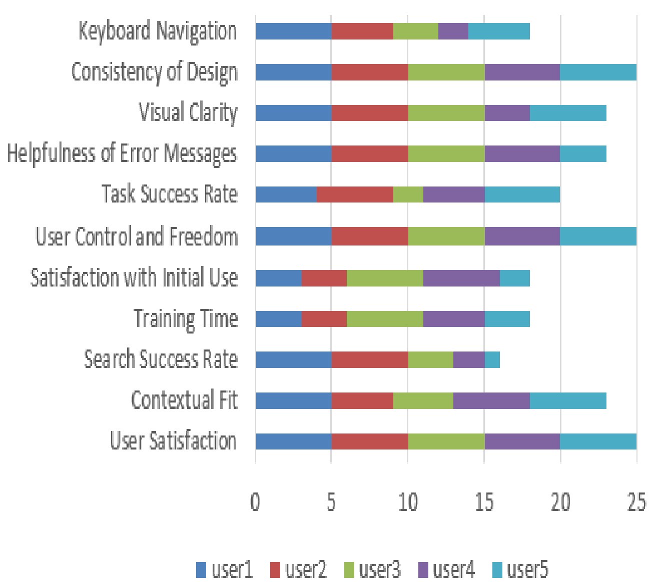

Figure 13. Usability assessment result.

5. Discussion

The next table shows the results of both quality attributes, which are combined. Here, the results of the existence of quality attributes are shown in the form of zero (0) if the system does not contain a certain attribute or one (1) if there exists a particular quality attribute in the system. The overall assessment of usability and maintainability is shown in Table 7.

Table 7. Results of quality assessment of 3D-ECSA.

	Maintainability					Usability						
	modularity	**reusability**	**Analyzability**	**modifiability**	**testability**	**appropriaten**	**recognisabilit**	**learnability**	**operability**	**user error**	**User interface**	**Accessibility**
Layered Architecture	1	1	1	1	1							
User interface						1	1	1	1	1	1	1

The gray area here is showing the lack of usability in layered architectural style as found from literature [3], [7]– [10] which we have attempted to achieve through interface design and also incorporated the advantages of 3D modeling in GUI.

The evaluation showed that layered architecture showed positive outcome for software maintainability for all of the measured attributes. Modularity, testability, reusability, modifiability, and analyzability were improved by this architecture, as it kept the concerns separated in an obvious manner. Isolation of layers helped improve modularity and testability, while reusable components and easier changes to individual layers facilitated reusability and modifiability. Analyzability was also improved since the system structure was clearer and easier to understand and optimize. However, over-layering or excessive abstraction can introduce unnecessary complexity, which may offset the benefits. Therefore, the balance between layering and simplicity is essential to keep the architecture effective.

It verified that the application of 3D modeling significantly increased the usability of software as for some properties. Appropriateness, accessibility and appropriateness got remarkable score because the data had been spatially visualized. Learnability was rather not damaged but was moderately negatively influenced by the threshold phase before adaptation to interacting features of 3D models. Operability was also positively affected, though this relied heavily on the design's simplicity and clarity. In particular, error protection during the interaction by users increased through 3D models offering clearer visual representation of data. Further improved interface aesthetics due to this dynamic presentation of a visual representation of rich 3D. Overall, the results suggest that 3D modeling, when properly designed, can significantly enhance the usability of software, particularly in the presentation of complex data. However, developers need to weigh the use of 3D visuals against considerations for accessibility and usability so that the interface remains user-friendly for all potential users.

6. Conclusion

This paper presents an implementation of a tool which was build using Layers architectural style and 3D modeling technique for immersive GUI in order to get the benefits of layered architecture upon the quality of software along with benefits of 3D interface for better user experience. We have developed the comprehensive system but for evaluation we assessed only two quality attributes including maintainability, usability. In future the detailed analysis comparison can be found for other quality attributes.

Users seem to like this new approach as they have deeper view over communication records rapidly through 3D representation. Because handling huge amount of contact records data in 2D graphs of simply tables with no graphical or interactive interface is time consuming and less attractive. And for maintainability we have found the modularity, reusability, modifiability, testability in our system by adding a new module of showing the persons over google maps which was incorporated later after building the other modules of the system.

Currently we got our system tested by general user not the targeted which include intelligence and defense agencies, mobile phone network companies. In future our system may be an interest of our targeted user to get their desired features which in long run may result in mitigating the crimes by using our system as a way to quickly identify the suspected person on the basis of their communication activities.

Funding: This research received no external funding.

Data Availability Statement: The datasets used during the current study are available from the corresponding author on reasonable request.

Ethical Statement: This research adheres to the ethical guidelines established by the Committee on Publication Ethics (COPE). All procedures and methodologies used in this study comply with COPE standards, ensuring transparency, integrity, and respect for all participants involved.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] Md. A. Al Imran, S. P. Lee, and M. A. Manazir Ahsan, “Quality driven architectural solutions selection approach through measuring impact factors,” in 2017 International Conference on Electrical Engineering and Computer Science (ICECOS), Palembang, Aug. 2017, pp. 131–136. doi: 10.1109/ICECOS.2017.8167119.

[2] M. Shaw and P. Clements, “The golden age of software architecture,” IEEE Software, vol. 23, no. 2, pp. 31–39, Mar. 2006, doi: 10.1109/MS.2006.58.

[3] N. B, N. Harrison, Paris, and P. Avgeriou, Leveraging Architecture Patterns to Satisfy Quality Attributes. 2007. doi: 10.1007/978-3-540-75132-8_21.

[4] L. Bass, P. Clements, and R. Kazman, Software architecture in practice, 3rd ed. Upper Saddle River, NJ: Addison-Wesley, 2013.

[5] “Queue PDF - Iso 25010 - Quality Model.pdf [3no7p9qmjeld].” https://idoc.pub/queue/iso-25010-quality-modelpdf-3no7p9qmjeld (accessed Nov. 06, 2022).

[6] C. Chen, R. Alfayez, K. Srisopha, B. Boehm, and L. Shi, “Why Is It Important to Measure Maintainability and What Are the Best Ways to Do It?,” in 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), 2017, pp. 377–378. doi: 10.1109/ICSE-C.2017.75.

[7] R. T. Fielding, “Software Architectural Styles for Network-based Applications,” p. 32.

[8] E. Majidi, M. Alemi, and H. Rashidi, “Software Architecture: A Survey and Classification,” in 2010 Second International Conference on Communication Software and Networks, 2010, pp. 454–460. doi: 10.1109/ICCSN.2010.94.

[9] N. Eftekhari, M. P. Rad, and H. Alinejad-Rokny, “Evaluation and Classifying Software Architecture Styles Due to Quality Attributes,” p. 6, 2011.

[10] F. Buschmann and K. Henney, “Pattern-oriented software architecture,” Jan. 1993.

[11] “Liveplasma Discovery Engine.” https://www.liveplasma.com/ (accessed Nov. 06, 2022).

[12] “AMiner - AI empowered science and technology intelligence mining.” https://www.aminer.org/ (accessed Nov. 22, 2022).

[13] J. P. Miguel, D. Mauricio, and G. Rodríguez, “A Review of Software Quality Models for the Evaluation of Software Products,” IJSEA, vol. 5, no. 6, pp. 31–53, Nov. 2014, doi: 10.5121/ijsea.2014.5603.

[14] “ISO 25010.” https://iso25000.com/index.php/en/iso-25000-standards/iso-25010 (accessed Nov. 22, 2022).

[15] R. N. Taylor and A. van der Hoek, “Software Design and Architecture The once and future focus of software engineering,” in Future of Software Engineering (FOSE ’07), Minneapolis, MN, May 2007, pp. 226–243. doi: 10.1109/FOSE.2007.21.

[16] D. Garlan and M. Shaw, “AN INTRODUCTION TO SOFTWARE ARCHITECTURE,” in Series on Software Engineering and Knowledge Engineering, vol. 2, WORLD SCIENTIFIC, 1993, pp. 1–39. doi: 10.1142/9789812798039_0001.

[17] Y. Viveros Martínez, E. López Domínguez, Y. Hernández Velázquez, S. Domínguez Isidro, M. A. Medina Nieto, and J. De La Calleja, “Layered Software Architecture for the Development of Third-Generation Video Surveillance Systems,” IEEE Access, vol. 7, pp. 98507–98521, 2019, doi: 10.1109/ACCESS.2019.2930401.

[18] L. A. Rivera Alvarado, E. López Domínguez, Y. Hernández Velázquez, S. Domínguez Isidro, and C. B. Excelente Toledo, “Layered Software Architecture for the Development of Mobile Learning Objects With Augmented Reality,” IEEE Access, vol. 6, pp. 57897–57909, 2018, doi: 10.1109/ACCESS.2018.2873976.

[19] H. Reza and E. Grant, “Quality-oriented software architecture,” in International Conference on Information Technology: Coding and Computing (ITCC’05) - Volume II, Las Vegas, NV, USA, 2005, pp. 140-145 Vol. 1. doi: 10.1109/ITCC.2005.237.