# SOFTWARE ARCHITECTURE DESIGNING CHALLENGES MODEL FOR INTERNET OF THINGS (IOT) SOFTWARE SYSTEM

**Mr. Noor Rehman[1], Dr. Abdul Wahid Khan[2]**

Department of Computer Science
University of Science & Technology, Bannu KPK, Pakistan.

## Abstract

Software Architecture is the vital part of every software to provide such mechanism which can enhance the rule of credibility in between client and vendor side organization. The IoT software will face with failure, if the architecture of it is not designed as desired. Elaborating the challenges faced by vendor side organization in the begging of the software development is an integral part of this study. The goal of Software Architecture designing Model (SADCM) for IoT software development will be used to reduce the failure chances of any software system in future use and to make a user friendly environment system. This will assist the vendor organization to overcome on all those challenges that create hurdles in designing of architecture for IoT software. Systematic review of the existing literature about this study will be analyzed to find the challenges. These challenges shall be validated by empirical approach and finally the proposed model shall be assessed by study in software Development organization.

**Key words:** *Software Architecture, Internet of Things, Software Architecture Patterns.*

**Corresponding author address:*

* Mr. Noor Rehman is the corresponding author of this paper. E-mail: noorkust301@gmail.com
* Dr. Abdul Wahid Khan supported in this research. E-mail: wahidkn@gmail.com

## 1. Introduction

Building the factual conclusion is the vital fragment of each efficacious software development and its assessment [1]. Many architecture model is already been designed which also plays a significant role in this concept, proposing such a model which can decrease the chances of failure of IoT software system in future is the main objective of this study. The IoT software outsourcing is the concept which deals with the process which takes place in between client and a vendor organization, where client proposed the required need and the vendor side organization fulfil this proposed requirement as desired in the system being in development stage. [2]. Software Architecture is the art of doing well-full designing of any software for the better performance as desired by the user during the development phase of software. Architecture of a software is just like a blueprint of the structure of a solid object [3]. Architecting the software is the general exercise of managing every origin and perimeter of software through an efficient way, just like a general prototype of a building before its actual construction which express that where would be the bath, kitchen and veranda etc in the home, so it is laid down on earth before the starting of actual work of construction on such layout. Architecture is the key success to overall better achievement of software in the whole cycle of its development. Software architecture conveys to descriptions that portray extraordinary level system design conclusions. In simple, software architecture is deals with the unrefined organization and worldwide control arrangement of a system, and goals to link the distance among the requirements stipulations and the execution of the system [3] . Software architecture enables early decision making capabilities about the system being developed. Internet of Things (IoT) the term internet of things has many ambiguous definitions. There are miscellaneous definitions for this term, it could be described as a system that connect the physical world with the digital components, computers, or automatic self-deriving vehicles. internet of things is a new and latest area of mixed software engineering and networking including artificial intelligence, which has brought out a new concept in the field of smart application development and ease for the beings in the community. It is assumed as a wideset of interconnected devices and machines to give significance to the daily life [2]. The mission critical system in which internet of things applications entail high existence, safety, security and reliability, serviceability, governing amenability and scalability. In addition in mission critical IoT system the architecture also causes its failure and scalability in case of un-proper development of the IoT software [3].

## 2. Literature Review

We have found through literature review that many studies dealing with this has own importance and significance to the scope of developing best architecture for IoT software. As architecture is the basic part of software development dealing with complex interaction with such environments to fulfil the requirements for IoT software in an efficient way and with user/client specified enlist objectives [3]. Software architecture is gaining supportable part and component for the efficient software development process for object-oriented applications [8]. Several patterns of software architecture are under use. **Layered Architecture pattern:** It is the most commonly used architecture pattern. Layered architecture concerned with the database and flow of data within the database. The data is passes through many layers (Tires) by splitting the code in layer and up to the bottom of each layer, every layer has a separate task in this pattern. Model view controller (MVC) is also the structure of layered architecture pattern. JAVA EE, Drupal, Express is the example models of Layered architecture pattern. Four layers are working in this pattern of architecture. Presentation layer is containing the graphical representation and design of the application. Business layer is dealing with the inclusion of model and logic for the specific business problem. Application layer sits in among presentation and business layer for providing abstraction. Persistence layer containing the code for accessing to database layer, it manipulates database Queries and statement etc [9]. **Advantages:** majority of the developer are more familiar with this pattern of architecture then others. This pattern of software architecture provides an easy way for writing well-defined/organized testable application [9]. **Disadvantages:** it inclines to lead to monolithic applications that are hard to be split up. The developer of layered application often find hard themselves for writing the code for this pattern due to passing a lot of code in each layer without adding any values in these layer [9].

**Micro-services software architecture:** it is a set of micro services which is actually for writing multiple applications that effort together. In this software architecture pattern micro-services has its own discrepancy responsibility and team can develop them autonomously like of other micro-services [10].

**Advantages:** it gives the accessibility to the developer to maintain, and arrange each micro-services unconnectedly. Micro-services software architecture is easier to be scaled. It is also easier to rewrite the pieces of the application due to smallest and less which is coupled in other parts [10]. **Disadvantages:** A single task or action of a user can pass through various micro-services that's why there are more chances of failure when something done to be wrong in the process. It is difficult to what you might to suppose, it is easier to write a code and then split it later into micro-services. A lot of extra concerns come into existence like communication, coordination, background compatibility etc [10].

**Event driven Software architecture pattern:** it is acting as a robotic system, it works when something happen in the form of action. It is very useful for a computer which interact directly with humans [11]. **Advantages:** it is easily adoptable to the complex system, often in confused environment. It can be scaled easily. It is easily extend-able in case of new event occurred [11]. **Disadvantages:** if the modules disturb each other than the testing process is difficult to be done and would be complex. Error handling is also difficult to structure [11].

**Blackboard Software Architecture pattern:** it is an architectural style related to artificial intelligence approach which can grip the difficult problems easily. It has a blackboard components which acts as a central data repository [12]. **Advantages:** blackboard Architecture style provides the con-currency which can allow knowledge sources to work in parallel. In this architecture style new application can be straightforwardly contracted through using the existing knowledge source [12]. **Disadvantages:** it has the provision of tight dependency in between the blackboard style and knowledge source. It is difficult to make a decision for cognitive termination. It is also lying an issue for synchronization of multiple agents [12].

## 3. Challenges

Software Architecture has many limitations expressed in many scenarios:

I. The first challenge to IoT software architecture is the nonexistence of gears and systematize methods to characterize software architecture [4].

II. Gears of investigation method to envisage that whether architecture will result in an enactment that sees the necessities or not [5].

III. The outsourcing issue which is awareness in among two parties vendor, client their poor understanding and poor expression [6].

IV. The gears of knowing of the design procedure, design knowledge and assessment of design [7].

## 4. **Five ages of Software Architecture**

As software architecture is the most pragmatic discipline in software development. The industry of software is progressed through its miscellaneous phases. Here we present the past five stages of software architecture and the action performed on each manner during the development [13].

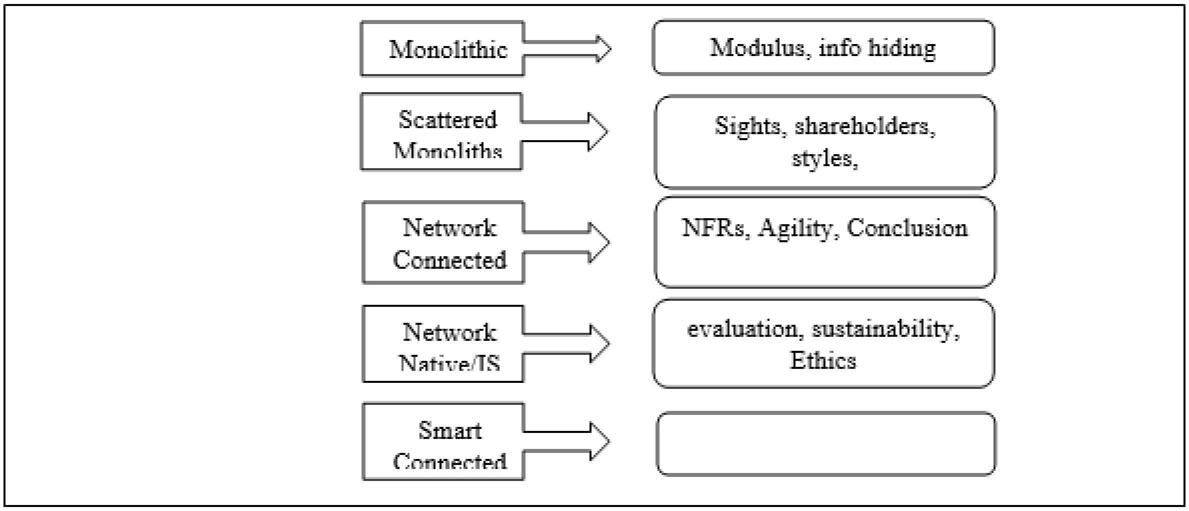

Fig.1: The assessment of Software Architecture.

The first stage of the software phases is monolithic which deals with the comprising of all in one slice. It is outmoded and integrated model of software program development [14].

As expertise are enthused to *distributed structures*, the snowballing complication of systems and their surroundings led to extra complicated design decision making and concerning long-term adjustments [15]. This led to software architecture actuality renowned as a discipline and the outline of methods to pact with these more complicated situation, such as lookouts and opinions for architectural explanation, clear documentation and administration of stakeholders [16]. The description of verified and recyclable architectural styles and the outline of architectural evaluation techniques to permit organized contrast of architectural options [17].*Internet connected* systems presented additional fresh defies that ensued in a fresh architectural attention on their defies non-functional potentials [18]. How to attain them and how architectural strategy could sustenance the suppleness desired to acquire alteration to market quickly and in a directive to answer to flying Internet-driven Souks [19]. In present period, *Internet-native or internet is the system of systems* which have ensued in additional alterations to the part of software architecture, as systems are nowadays often far more soft and vibrant, being collected from fine-grained system amenities (micro services). Systems are often erected on PaaS (platform as a service) stands and so can be collected from a mixture of platform with services. This has destined that the software architect must to be apprehensive with empowering fast and dependable development of the system. Thus, the architecture is supportable over the long term, and the Architecture would be distinct more as a usual of decorations and moralities rather than a motionless structure that leftovers steady for an extended era. As we aspect to the upcoming and Intelligent Connected systems development that how will the character of the software architect change again?

## 5. Problem Statement

In the early developed models of architecture for software is good enough at their end. For today digital world the scope and the neediness of such smart applications which could fulfil the actual task of every human and interact with the system has their own advantage. The main role of every project successful completion is the vendor side organization where they should focus to identify the challenges that create hurdles in designing of software architecture for IoT software development. Data flow constraint occurred in every un-proper developed software system which causes failure of information of that system. Dealing with data and giving focus to data flow in designing has their own smartness.

Following limitations have been found in the existing Models/Patterns of Software Architecture.

- Derive architecture that will address non-functional requirements
- Relationship of connectors to middleware
- Variables aren't persistent nearly as much as they should be.
- Too much reliance on external databases.
- UI is still a messy hack.
- Cross-platform is still take too much effort.
- Exceptions are still handled poorly.
- Return types don't carry any status by default.
- The outstanding issues are the potential longevity of the software and the versatility to add features desired.
- GUI is hard.
- Automatic parallelization is impossible.
- Dynamic typed languages are slow and there's a limit on how fast we can make them.
- Garbage collection produces long, hard pauses.

## 6. Significance of the work

Software architecture designing challenges model (SADCM) will be a new model from vendor perspective that will highlight all those challenges that create hurdles in designing a new architecture for IoT software development. This model will support the software outsourcing vendor organization to assess their level for designing architecture for IoT software development. This Model will explore the limitations in the existing models/patterns of Architecture models and will propose a solution to each challenge/limitation of software architecture model.

## 7. Research Objective

Research objectives for the development of Software Architecture designing Challenges Model are:-

- To conduct Systematic Literature review for the documentation of Limitation.
- To Establish SLR for documentation of exercises for the recognized limitations and challenges.
- To Establish Questionnaire Survey for the authentication of SLR findings and to find some latest challenges/practices apart from the recognized ones.
- To establish case study for the validation of proposed model.

## 8. Research Question's

**Q.1:** what are the challenges faced by software outsourcing vendor organization for software architecture designing for IoT.
**Q.2:** what are the practices to handle the identified challenges by software outsourcing vendor organization for designing software architecture for IoT.
**Q.3:** what are the real world practices to handle the challenges by software outsourcing vendor organization for designing software architecture for IoT.

We elaborate **Q1** to look into the challenges faced by client and vender side organization in the development of software architecture. Moreover, we will also define/elaborate **Q2** for the acquired practices faced by vendor organization in software outsourcing. Finally, in **Q3** we will refer a solution for the successful development of software architecture in real world environment.

## 9. Research Methodology

Research methodology will start by reviewing the existing knowledge and information about the in-process study. These all knowledge would be achieved through systematic review and will be analyzed for further findings and gaps. All the primary and secondary data will be the part of designing such architecture model for IoT software.

### 9.1 Systematic Literature Review

Systematic literature review which is often known as a systematic review is deals with identification, evaluation and interpretation of all available research stuff for a specific research question, or research area [2]. Systematic literature review is the guidelines of how to summarize the existence stuff for the treatment of specific research question [6]. Systematic literature review is also a type of empirical method, which aims to investigate a specific research questions. Systematic literature review will cover Q1 for data collection and for the identification of related research work through analysis and data gathering practices [2].

### 9.2 Empirical Method

Empirical engineering is the process of analyzing, interpreting the existence stuff for a specific research question. Empirical process has the main focus on gathering the data relevant to the specific research question as an evidence through experiments and measurements [20].

### 9.3 Case Study

Our study found Case Study as the most important longitudinal study for the collection of data for our study. In the sense of internet of things major software architecture issues has been undertaken and a deep analysis of these issues is needed for consideration. These will represent a general justification for the conductance of our case study [3].

## 10 Research Plan

We have intended to gather the data through SLR in the first phase. Secondly, the SLR conclusion will be authenticated the questionnaire survey in local and abroad software outsourcing industry. Third, on the basis of SLR and QS findings we will define various levels of our proposed model. Fourth, the proposed model will be assessed through case study in software subcontracting industry to get feedback about its reliability and changes will be incorporated as suggested by the experts of industry.

## 11 SADCM Model Development

In the entire cycle of development SADCM total five stages were followed for the successful development of the above proposed model. In the first stage of the development SADCM, it will specify the measures for its successful development.

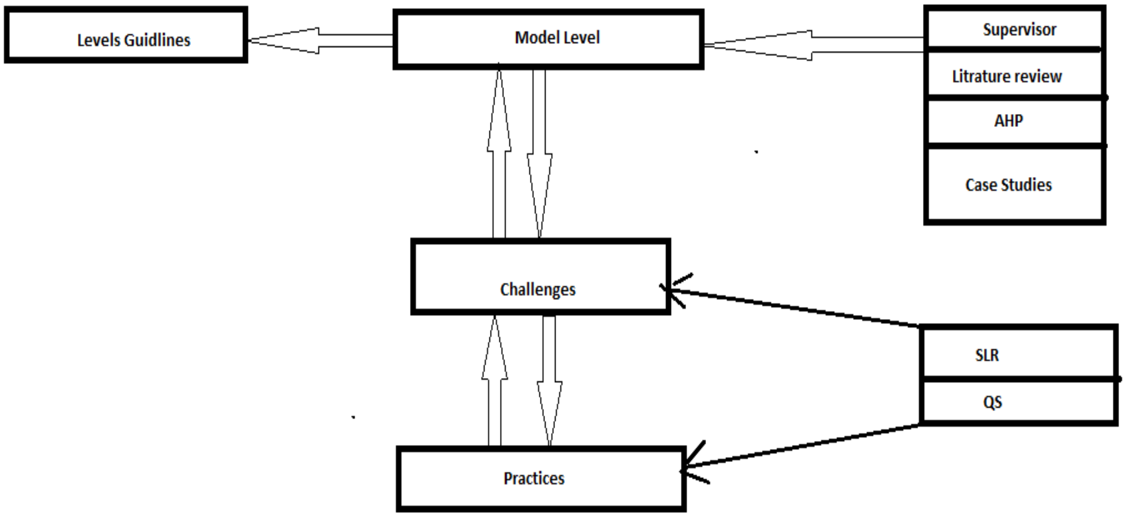

Fig: 2. Proposed model for SADCM

The ensure development and measurement criteria comes from the review of relevant literature [1-2-5]. The achieved criteria for the deliberation would be as follows:

- **User Satisfaction:** outsourcing vendor will be satisfied from the result of the Software Architecture Designing Challenges Model for the anticipated outcomes relevant to theirs prerequisite without any hesitancy.
- **Ease of Use:** the structure of the proposed model will be easy for understanding and interaction of the user with the developed model.

In the second stage the data collection process will be done through SLR and received data will be validated through questionnaire survey in the industry of outsourcing. In the stage three the structuring and rationalization of the result would be done, which will be grounded on empirical results. The four stage will express the outcomes achieved from the previous three stages. In the final stage of this model will be the initialization of evaluation stage, which will be cover through case studies in order to assess the applicability of the proposed model for software architecture.

## 12 ACKNOWLEDGEMENTS

The authors are very much obliged to University of Science & Technology, Bannu for the interminable support and alliances.

## 13 References

1. Naab, M.R., Dominik, How to Evaluate Software Architecture, in 2018 IEEE International Conference on Software Architecture Companion. 2018, IEEE Xplore: Germoney. p. 2.
2. khan, A.W., Khan, Sifat ullah. Outsourcing Contract Management Model (OCMM). in Madrid. 2012. Spain.
3. Stavropoulou Ioanna, G.M., Kontogiannis Kostas Case study on which relations to use for clustering-based software architecture recovery, CrossMark, Editor. 2017: New York. p. 46.
4. Uslar Mathias, R.S., Neureiter christian, Pröstl Andrén Filip, Velasquez **Jorge**, Applying the Smart Grid Architecture Model for Designing and Validating System-of-Systems in the Power and Energy Domain: A European Perspective. MDPI, 2019: p. 40.
5. Kaur Parmindar, S.H., A Case Study Based Approach for Effective Use of Software Architecture Guidelines. International Journal of Computer Applications, 2012: p. 6.
6. Clerc Viktor, P.L., van Vliet Hans, The Architect's Mindset. SpringerLink, 2007(4880, pp. 231–249, ): p. 19.
7. Gericke Kilian, B.l., An analysis of design process models across disciplines. ResearchGate, 2012: p. 11.
8. Chabbani, M., Rodriguez, Ismail, Drira, Khalil, Jmail Mohammed, Mining Approach for Software Architectures' Description Discovery, in 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications. 2017: France. p. 8.
9. Belle boay Alvine, M.M., The Layered Architecture revisited: Is it an Optimization Problem? ResearchGate, 2013: p. 7.
10. Marquez gaston, A.H., Actual Use of Architectural Patterns in Microservices-based Open Source Projects. December, 2018: p. 11.
11. Vettaleti A, F., Memeo and Landi, Pucci.p and Laura.G, Smart City: An Event Driven Architecture for Monitoring Public Spaces with Heterogeneous Sensors. ResearchGate, July, 2010: p. 7.
12. Stegemann Kleine, s., Funk Burkhardt A Blackboard Architecture for Workflows. ResearchGate, 2007: p. 5.
13. eion, w., Software Architecture in a Changing World. ResearchGate, November, 2016: p. 7.
14. D.L, P., On the Criteria To Be sed in Decomposing Systems into Modules. ACM, 1972. **15**: p. 6.
15. ISO, I., IEEE, 42010, Systems and software engineering —Architecture description. IEEE Xplore, 2011: p. 46.

16. Garlan, d.s.m., An Introduction to Software Architecture. IEEE Xplore, 1994: p. 42.
17. Paul, c.R., Kazma Klien, Mark addison-wesley, Evaluating software architectures: methods and case studies. ACM, 2002: p. 3.
18. Woods, E.R.N., Using Architectural Perspectives. IEEE Xplore: p. 10.
19. Abrahamsson, p.a., muhammad babar, Agility and Architecture: Can They Coexist? IEEE Xplore, 2010: p. 13.
20. Galster Mathias, w.d., Tang Antony, Kazman Rick, From Craft to Science: The Road Ahead for Empirical Software Engineering Research. ACM/IEEE, 2018: p. 4.