

Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science

Saidur Rahman

**Comparative analysis about the challenges and implications of IoT
development methodologies**

Examiners : Associate Professor Uolevi Nikula
Post-doctoral Researcher Sami Jantunen

Supervisors: Post-doctoral Researcher Sami Jantunen
Associate Professor Uolevi Nikula

ABSTRACT

Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science

Saidur Rahman

Comparative analysis about the challenges and implications of IoT development methodologies

Master's Thesis
2018

55 pages, 5 figures, 7 tables, 2 appendix

Examiners: Associate Professor Uolevi Nikula
Post-doctoral Researcher Sami Jantunen

Keywords: IoT, SLR, IoT software development

The expanding nature of Internet of Things is expected to outbreak the future of internet. Devices connecting together to form an IoT ecosystem can bring revolutionary changes in human lives that can introduce numerous research opportunities. The diversified nature of an IoT application can be a challenging issue for the software development industry. Threats like security, interoperability etc. related to the IoT application can be more difficult to handle that can bring some implications in the software development methods. The increasing number of challenges and implications in IoT application can change the steps in software development process. Moreover, the existing best practices of software development methods are limited to their scope to handle the challenges and implications properly. Therefore, the motive of this research study is to find out the challenges and implications of IoT software development by conducting a systematic literature review.

Also, analyzing the existing methods of IoT software development to sort out on what extent they are handling the identified challenges and implications.

ACKNOWLEDGEMENTS

First of all, I am grateful to almighty Allah that he has blessed me with a chance to come in Finland. I am enthusiastically urged to write some words about the people who have strained my mind with high influential remarks here. The biggest motivation I have found is from my parents whose arduous efforts has made my dream true and without them it was almost impossible to fulfill my goal. I am also grateful to my brother, sister, friends and all my family members who have been a great support for me from the very beginning of my life.

Apart from my family and friends, the people I am really grateful to are the teachers from Lappeenranta University Technology. I have learned a lot from them and the journey towards my master's degree with them was quite inspiring. Specially, I would like to thank my co-supervisor Mr. Uolevi Nikula who gave me the opportunity to work with him with the purpose to write my master's thesis. Mr. Uolevi is one of the best professors from Lappeenranta University of Technology from whom I have learnt lots of new things. The name that I am bound to mention and have always to remember till my last breath is Mr. Sami Jantunen, my supervisor. Mr. Sami was amazingly co-operative to me and from the very beginning his scholarly guidance helped me a lot to complete my thesis. I have learnt a lot from him specially, the process of conducting a thesis in practice and without him it would not be as easier as it was so far.

Last but not the least, I am grateful to the people from Lappeenranta, my classmates who came across with me and gave me enough company to keep my concentration sticked to study. Once again, I am very much overjoyed that finally I have concluded my thesis and am looking forward to contribute more in the research area of software engineering.

Author

Saidur Rahman

25/05/2018

TABLE OF CONTENTS

1 INTRODUCTION	4
1.1BACKGROUND.....	4
1.2GOALS AND DELIMITATIONS	6
1.3STRUCTURE OF THE THESIS	8
2 RESEARCH METHODOLOGY.....	9
2.1SYSTEMATIC LITERATURE REVIEW.....	9
2.2 REVIEW PROCESS.....	10
3 RESEARCH DESIGN.....	14
3.1 DEVELOP AND VALIDATE THE RESEARCH PROTOCOL.....	14
3.2 FORMULATE THE RESEARCH QUESTIONS	15
3.3 SEARCHING PROCEDURE	16
3.4 INCLUSION AND EXCLUSION CRITERIA	19
3.5 QUALITY ASSESSMENT	19
3.6 PRIMARY STUDY SELECTION	20
3.7 DATA EXTRACTION.....	21
4 DATA SYNTHESIS & RESULTS.....	22
4.1 RESEARCH QUESTION 1.....	22
4.2 RESEARCH QUESTION 2.....	30
5 DISCUSSIONS AND CONCLUSIONS	45
REFERENCES	47
APPENDICES	
APPENDIX 1	
APPENDIX 2	

LIST OF TABLES

Table 1: Research Questions.....	7
Table 2: Summary of PIO Criteria.....	12
Table 3: Electronic Sources for SLR	17
Table 4: Quality Assessment Checklists.....	19
Table 5: Data Collection	21
Table 6: Ignite IoT Development Methodology	31
Table 7: Comparisons of IoT development methodologies.....	43

LIST OF FIGURES

Figure 1: Systematic Literature Review	11
Figure 2: Selection of primary studies	20
Figure 3: IoT related challenges	23
Figure 4: IoT related implications	28
Figure 5: The IoT Methodology	33

LIST OF SYMBOLS AND ABBREVIATIONS

IoT	Internet of Things
SLR	Systematic Literature Review
RFID	Rapid Frequency Identification Number
WSN	Wireless Sensors Network
GPS	Global Positioning System
API	Application Programming Interface
TLS	Transport Layer Security
SAS	Shared Access Signature
CI	Continuous Integration
CD	Continuous Development
QA	Quality Assurance
AIA	Asset Integration Architecture
OSI	Open System Interconnection
MVP	Minimum Viable Product
LED	Light-emitting diode
TCP	Transmission Control Protocol
IP	Internet Protocol
REST	Representational State Transfer
UX	User Experience
PD	Project Dimension
Wi-Fi	Wireless Fidelity

1 INTRODUCTION

1.1 Background

IoT is the network of things that is turning the world into a new era where billions of objects can sense, communicate and share information (Yeo et al., 2014). These interconnected devices share and exchange data to initiate action, provide intelligence for planning, management and decision making. IoT creates a new way to access the information anytime from anywhere, share the information of things real time and accurately, analyze and process the vast amount of data through different intelligent computing devices and reliable transmissions networks (Liu and Lu, 2012). In general, IoT is the technology that combines machine devices with internet devices where a computer receives data from all connected things to process and monitor data and create services for the society. For example, a connected car is a vehicle which is able to optimize its own functions, maintenance and provide comfort for the passenger by using onboard sensing and internet connectivity (“10 Real World Applications of Internet of Things (IoT) - Explained in Videos,” 2016). From smart home to smart city the application of IoT is expanding rapidly. IoT allows its users to achieve deeper automation, analysis and combination within the system.

The concept of IoT became popular in 1999 through Auto-ID center at MIT(Massachusetts Institute of Technology) (“History of IoT | Background Information and Timeline of the Trending Topic,” 2016). From then till now, the term became the driving force of technological advancement. Different companies, individuals, academic institutions have made a significant role to implement IoT solution for their needs. To develop IoT solution different technologies needs to work together. The key technologies of IoT includes RFID, Sensors, WSN networks, GPS, Bluetooth, Zig Bee, internet and mobile networks, network services, industry applications that has been used to various purposes such as: data acquisitions, signal processing, network access, design verifications, intelligent information processing, information integration and support and application and other areas (Liu and Lu, 2012; Liu and Zhou, 2012).

These are the most common technologies that work together in IoT to connect different physical medium such as: home, car, mobile, laptops and so on. It has been estimated that the number of physical medium connects with IoT will exceed 28 billion by the year 2020

(Hua, 2016a). The expanding number of connected IoT devices will provide numerous challenges for IoT development. For example, IoT application runs on a network that consists of heterogenous devices. These devices vary in terms of their characteristic types such as: sensors, actuators, RFID tags, user interfaces, storages, interaction modes (periodic, event driven, request-response, notify) and different platforms (Android, IOS, Linux, micro-controller with no operating system). The heterogenous nature of IoT application brings challenges for IoT application developers to address issues that are related to specific development life cycle including design, implementation and deployment. For example, during the design phase the application logic needs to be understand and several tasks should be allocated for various types of heterogenous devices. These tasks then need to be implemented for different platforms and then finally application has to be deployed into a network of devices (Chauhan et al., 2016). The underlying phases of IoT application development prove that it needs sheer determination of manual effort which can be time consuming as well as error prone.

As IoT is evolving and expanding to billions of devices adoption of IoT will bring multiple barriers including security, trust and privacy, integrations and testing, maintenance and changing requirements etc (Kocher and Heron, 2014). These barriers will create new business opportunities for technology companies, middleware developers, system integrators and cross platform builders. IoT connects different devices together that can provide decentralized entry points for malware that can be vulnerable to security issues. The core use case of IoT is remote sensors and monitoring can create difficulties with issues such as: trust and privacy and controlling and ownership of data. The different platforms, numerous amounts of protocols and large number of API can create complexities with IoT integration and testing. The distributed nature of IoT application can be challenging to maintain and changing the requirements due to the ecosystem of IoT development life cycle where multiple devices needs to control simultaneously.

Furthermore, developing and maintaining the complex IoT system is challenging due to the parallel development of multiple resources that need to operate together. For example, there can be multiple teams, organizations, vendors interacting together to develop an IoT solution. Combining and integrating them depends on several issues such as: security, privacy, handling the data etc. which increases the amount to be worked more in IoT ecosystem. Otherwise, the development of IoT can be resulted in project failure. The

success of IoT development largely depends on handling different components and how they will interact with each other. Additionally, 75% of IoT software and hardware development projects failed to meet the actual business objectives due to the lack of collaboration of different departments and the lack of innovations in the software development approach (“IoT Development Challenges,” 2017). Also, a recent survey explains that nearly 34% of Internet of Things initiatives are considered as a failure, while a third of all projects being completed were not seen as a success (“Internet of Things project failure and success 2017,” 2017). As the organizations are emerging with the new trend of IoT, the challenges regarding IoT development will increase frequently. Therefore, investigating and conceptualization of IoT and the IoT development trends is a significant part for scientific research.

1.2 Goals and delimitations

The number of endpoints in the IoT application will emerge as the technology advances and the ratio of challenges with IoT development will be in the rising manner. It can bring complexities with the IoT development approaches. The challenges with IoT development will not only be limited with the well-established areas like security, safety, reliability but also with the interaction between those areas. Heterogeneity, IoT ecosystem, the new types of connections, configuration settings and changing capabilities will alter the laws and definitions of software development. IoT applications are distributed compared to other software systems, as a result new failures mode due to communications, exception condition management and reliability engineering will come into play for the IoT development (Jacobson et al., 2017).

Yet, best approaches of IoT software development is still emerging and suitable test and validations environment are in their infancy (Favaro, 2015). Besides, existing traditional software development method is not capable enough to handle the issues related with IoT development method. Traditional software development method such as: waterfall, agile, practices such as scrum, RUP(Rational Unified Process) often need more time and cost for IoT development and it is less constructive for handling the data security, get-way priority and managing the changing of requirements in the IoT development lifecycle (“Problems with Traditional Methods of Software Development,” 2013).

To address the challenges that discussed above various approaches of IoT development

have been proposed. Node-centric programming is one of them that has complete control over individual devices for developing extremely efficient systems (Grace et al., 2016a). However, it is not easy to use for IoT application development due to the heterogeneity and large size of system. There are some other models implemented to handle the challenges for IoT development such as: IoT Ignite, IoT Methodology, IBM garage. These methodologies have been used broadly for IoT development in the recent years. These methodologies are monolithic methods that reuse many existing practices, combining these with new innovative practices especially with IoT is challenging (Jacobson et al., 2017). Also, to adopt those methodologies for a domain specific problem might be difficult in nature for the industry.

Therefore, it is an important fact to investigate more about the area to address the research challenges behind this technology. The goal of this current research study is to investigate more about IoT development problems and how it changes the approaches of software development method. Exploring and analyzing different existing IoT development methodology to understand how they are dealing with the current IoT software development problems.

Table 1: Research Questions

Research Questions		Goal
RQ1: How the software developments have changed because of IoT?	RQ1.1: How the characteristic of software development problems is changing because of IoT?	Identifying and understanding the current problems with IoT development and finding out the challenges and implications that the existing development methodologies are dealing with.
	RQ1.2: What implications identified IoT-related characteristics pose on software development methodologies?	

RQ2: What IoT development methods have been proposed?	RQ2.1: How the proposed methodologies are dealing with current challenges of IoT development?	Finding the most common IoT development approaches and explore the differences between them.
	RQ2.2: What are the limitations, similarities and differences between those methodologies?	

1.3 Structure of the thesis

The structure of the thesis is categorized into five chapters. Chapter 1 provides the information about research background, goals and delimitations and structure of the thesis. Chapter 2 explains the methodology and the process to be followed for this current research study. Chapter 3 explains the whole research design of the thesis. Chapter 4 demonstrates the arguments, study findings from the research study in a consistent manner. Chapter 5 explores the discussions and conclusions of this research study.

2 RESEARCH METHODOLOGY

As, the primary purpose of this research study is to develop the general understanding about IoT related challenges and finding out the research gaps, I consider to explore the research topic with systematic literature review. I believe SLR would be the appropriate research methodology for this particular research study because of its acceptability for the exploration, analysis and synthesis of a research area in a systematic way (Kitchenham et al., 2009). Moreover, the purpose of our research study is to gain in depth knowledge about the phenomena to understand the problem areas more where the SLR can be a suitable option. SLR defines specific search criteria, quality assessment and resources to be used before conducting the actual research which helps the researcher to avoid bias during the actual research study. According to Kitchenham (2007) SLR is best suited where researchers have to summarize the existing evidence, to identify the gaps in current process and to provide a new frameworks or models for the phenomena of interest. Overall, the systematic way of conducting research through SLR that can satisfy my research needs for this particular study motivates me to choose SLR as a research method.

2.1 Systematic Literature Review

The Systematic literature review helps researcher to study, understand and to gain insight knowledge about the phenomena. According to Kitchenham (2009): “A systematic literature review (often referred to as a systematic review) is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest”.

There are some fundamental benefits of Systematic literature review in research. It helps to identify the specific research needs to be identified and to formulate the research questions in practice. Besides, this approach of research performs well to find out the targeted research group, people, intervention and finding out the research gap from existing research articles. Moreover, it helps researcher to summarize, think broadly about the concept of the research. Also, it defines a method to validate the research findings and outcomes (Sulayman and Mendes, 2009).

Systematic literature review is not similar like normal literature review. There are certain things that differentiate systematic literature review from conventional literature review.

Systematic literature review is well organized in structure which starts by designing the review protocol to be followed during the whole research study. Research questions, objectives and the methods are addressed that will be used to perform the review. It defines a specific search strategy, data collection and synthesis process to satisfy the research needs. Also, systematic literature review requires the inclusion and exclusion criteria for each research study and to ensure the quality of the study, it defines the quality assessment process to be used during the whole review process. According to Kitchenham writing the review is one of the fundamental features of systematic literature review to allow the reader easily access the research findings (2004.).

2.2 Review Process

The research review protocol is the iterative process that contains the overall process of the whole systematic literature review. Therefore, the steps of review protocol to be followed during this current research studies are illustrated below:



Figure 1: Systematic Literature Review

Planning

Need of a Systematic Review: This phase is essential to develop the overall understanding about SLR. There can be such situations that SLR is not necessary. Before, conducting the actual research the researcher ensures themselves that SLR is required for a particular research study. In this phase, the reasons and the proper explanation why it is needed to do a SLR for our current research study would be explained and discussed.

The current research objective is to explore the existing IoT development methods to find out the problems, reasons why IoT development is still a challenging task. To satisfy the research needs SLR can be a suitable options to follow as a guidelines for this research

study where the current trends, problems of IoT development methodology and the limitations, similarities and differences will be sorted out.

Formulate the Research Questions: Formulation of the research questions is an important stage in SLR based on the article of Kitchenham (2009). This is an iterative phase where the important research questions will be investigated and formulated for this study. Kitchenham used three criteria to frame the research questions in more generic way (2009). The criteria include Population, Intervention and Output.

To formulate our questions structure we have used the three criteria defined by Kitchenman et al. as described in Table 2:

Table 2: Summary of PIO Criteria

Population	Individuals, Industries, Software engineering researchers whose focus is to know about the challenges of IoT development methodology
Intervention	Various practices of IoT development and their methodology such as: Ignite, IBM garage etc.
Output	Identifying the underlying challenges of IoT development methodology

Develop and validate the review protocol: This phase is also iterative and covers the overall planning of the SLR. Review protocol is necessary to avoid researchers' bias and it acts as an agreement between the researchers. The review protocol for our study will be discovered step by step in chapter 3.

Conducting

Searching process: Searching process in SLR is an indispensable part as it defines the overall searching criteria to be used the SLR. The searching strategy will be performed by using the related key words of the research topic. In general, in this phase the resources (journals, articles, conferences etc) to be used during the SLR will be identified.

Inclusion and Exclusion criteria: During this phase the criteria to be used for this particular SLR will be implemented. The criteria will define which study materials should

be included and which one will be excluded from the study. Inclusion and Exclusion criteria should be defined based on the research questions so that it classifies the RQ correctly.

Quality Assessment: A quality checklist is needed to assess the quality of the SLR. It helps the researchers provide guidance for further research and to ensure that the current research study is enough to answers the RQ correctly. Therefore, in this phase a quality checklist will be designed to be followed for this particular research study.

Selection of primary study: While doing SLR it is important to show the details about the selected study materials. Therefore, in this phase of SLR the number of final selected articles will be illustrated

Data Extraction: In this phase of SLR the relevant data related to the RQ will be extracted from each study. This is a continuous process to be fulfilled until the enough data collected. The objective of data extraction is to obtain the information from the study needed to address the RQ. Additional information regarding RQ can be collected while designing the data extraction form.

Data Synthesis: Once the data extraction is done, the data will be investigated more to clarify, analyze and summarize to answers the RQ. Data synthesis can be descriptive in nature. It can be designed in tabulated manner to show the result from the study.

Documenting

Writing the report: This is a single step where the actual report for the particular SLR will be written in details to demonstrate the findings from the research. In this part of research the whole findings from this SLR will be written in the master's thesis format provided by Lappeenranta University of Technology.

3 RESEARCH DESIGN

3.1 Develop and Validate the Research Protocol

Initially, I have started my research by doing some background study about the IoT development methods. From the background study, I have found that there are enormous challenges regarding the topic area such as: security, interoperability, test automation and ensuring the quality of services, risk handling etc. and equally current approaches are still in their infancy to mitigate those IoT challenges (Favaro, 2015; Hua, 2016b; Yeo et al., 2014). I wanted to have some in depth knowledge about the phenomena to understand the problem areas and decided to do a SLR based on the topic. I have decided to follow the guidelines provided by Kitchenham (2009) to develop the review protocol for this research study. At the beginning of my SLR I tried to sort out the reasons to do SLR for this topic. As, the study advanced I tried to formulate the tentative research questions based on my research purpose. Formulation of the research questions was an iterative manner whereas several changes happened to meet the actual research needs. Both the research questions were formulated in way so that the research answers can be enough to satisfy the research objectives. To find the relevant articles for the research questions a searching procedure is formed for this study. The searching procedure defines the sources to be used for this study. I have selected six sources to conduct the searching strategy of my research. Also, if needed Google search with the specific key words will be performed. Those resources were selected based on their fame and most usage references for other articles during the years. The searching process will be followed up by several inclusion and exclusion criteria to find the exact materials for the research questions. For instance, the articles that focused on IoT development methods will be included directly to the study process whereas there can be some articles that are not relevant with our objectives and purpose will be excluded from the study. Furthermore, the protocol of this study defines quality assessment criteria to check if the purpose of the research study is focused or not and whether the research answers from the study materials are enough to answers all the research questions for this topic. There are several data items were chosen to collect the data from study materials so that the answers of research questions can be found. The data will be collected and stored to a spreadsheet for further analysis. Once enough data will be collected the analysis part will begin to answers the research questions.

The review protocol for this research study will be incrementally illustrated in the following manner. The supervisor and one experienced Post-doctoral researcher will evaluate the review protocol. The protocol will be changed and updated with several iterations based on the recommendation of supervisor and the post-doctoral researcher.

3.2 Formulate the research questions

The current research problem area is to understand the difficulties with IoT development methodology and identifying why the existing methods are still challenging to solve those problems. Therefore, the research questions are formulated based on that needs. There were several iterations to form the research questions to match those with the research objectives.

Initially, I have formulated the research questions into two categories where RQ1 was “How the software development have changed because of IoT ” and RQ2 was “What IoT development have been proposed ”. To simplify the research questions more both RQ’s were categorized into two sub questions. Following is the initial version of research questions that I came up for this current research study.

RQ1: How the software development changed because of IoT?

RQ1.1: What types of problems IoT is introducing to software development?

RQ1.2:What are the challenges that current methodologies are insufficient for dealing with such problems?

RQ2: What IoT development methods have been proposed?

RQ2.1: How the proposed methodologies are dealing with current challenges of IoT development?

RQ2.2:What are the limitations, similarities and differences between those methodologies?

To clarify the RQ’s more towards the research objectives; the initial version of RQ’s analyzed carefully. After careful investigations, I have found a dilemma between the RQ1.2 and RQ2.1. Both were seems similar in objectives and nature. Therefore, I had to change the RQ1 to clarify the RQ’s more and finally I came up with the following RQ’s.

RQ1: How the software developments have changed because of IoT?

RQ1.1: How the characteristic of software development problems is changing because

of IoT?

RQ1.2: What implications identified IoT-related characteristics pose on software development methodologies?

RQ2: What IoT development methods have been proposed?

RQ2.1: How the proposed methodologies are dealing with current challenges of IoT development?

RQ2.2: What are the limitations, similarities and differences between those methodologies?

The RQ1 will focus on the current trends, problem areas, challenges of IoT software development. On the other hand, the RQ2 will help to find out the existing methods of IoT development and it will help to explore more why the existing methods are insufficient to deal with the current challenges of IoT development.

3.3 Searching Procedure

The primary searching process is to retrieve the relevant literature will depend on the Online Database, Surfing through Search Engines, Journals, Conferences, Master's or PHD dissertations from the various sources.

The searching procedure was followed by using some key search terms based on the research questions. The searching terms that were initially used named as: **Internet of Things, IoT development methods, Challenges with IoT development methods, IoT development methodology, IoT software development method**. In addition singular or plural forms of those keywords were considered while searching and those terms were connected with different logical operators such as: **AND, OR** depending on the variations of searching needs. Using the initial version of the key words it was not enough to find the suitable articles for this research study. Therefore, additionally some other key words such as: **IoT software development process, challenges with IoT software process models, current approaches of IoT software development, existing IoT software development methods** were included in the search terms.

In order to select the most suitable databases and resources for the search terms the several criteria were considered. The criterion includes content update: if the content of the articles are regularly updated, availability: if the contents are available to use in scientific community, accuracy of the results obtained by the search (Maia et al., 2014).

Based on the criteria discussed above several electronic sources were selected, as shown in Table 3. The lists of the selected study resources are given below:

Table 3: Electronic Sources for SLR

Name	URL
IEEE Xplore	http://ieeexplore.ieee.org
ACM digital library	http://dl.acm.org
Springer Link	http://link.springer.com
Science Direct	http://www.sciencedirect.com
ISI web of Services	http://www.webofknowledge.com
LUTPub/Doria(LuT)	http://www.doria.fi/handle/10024/4000
Google Search	https://www.google.com/

The database sources selected in Table 3 are known as most commonly used database sources for Systematic Literature Review in Software engineering as defined by Kitchenham and Maia (2007; 2014.). Also, the Master's or PHD related thesis article from Lappeenranta University of Technology Library and independent search query through Google were considered to find some relevant articles, websites and resources.

The selected resources were chosen because they have been using for another research study by the top institution and top scholar in the field of software engineering over the years (Glass, 1994). Also, the availability of those resources through LUT Online library was considered while selecting.

The final version of the selected study sources shown in Table 3 are refined form of initial version of source materials. The initial versions of selected resources were formulated in the following manner.

Online Database

- IEEE Xplore
- ACM digital library
- Springer Link
- Science Direct
- LUTPub/Doria(LUT)

Journals

- Journal of Systems and Software (JSS)
- Empirical Software Engineering (EMSE)
- IEEE Transactions on Software Engineering (TSE)
- Journal of Software Process Improvement and Practice
- International Journal of Internet of Things in Cyber Assurance
- International journal of Internet of Things

Conferences

- International Conference on Software Engineering (ICSE)
- EuroSPI (European Conferences on Software Process Improvement)
- International SPICE Conferences on Process Assessments and Improvement
- IoT Evolution Expo
- IoT International Innovation Conference

Dissertations

- Master's or PHD related thesis article
- Lappeenranta University of Technology Library

Web Sites

- Independent search query using Google

Although the initial version of resources seems well organized and categorized but there were some complexities following that structure. First of all, finding relevant literature from each and specific resources using the searching key words defined before were time consuming. In some cases, it was confusing to which items should be selected first to find the most appropriate article for the study. Additionally, some of the listing resources such as: International journal of Internet of Things and International Journal of Internet of Things in Cyber Assurance were not enough to find the relevant article related to this research study due to the unavailability of free access to the materials. Therefore, the formulated resources in Table 3 were most suitable and comprehensive to satisfy the research needs.

3.4 Inclusion and Exclusion Criteria

Inclusion and exclusion criteria define what types of resources will be followed during the Systematic literature review and what types of resources will be omitted from the review process. Below are the terms that will be followed as Inclusion and Exclusion criteria for this current research study.

Inclusion Criteria:

- Scientific paper that are being focused with IoT and IoT development methodologies
- Article with full text
- Paper with clearly highlighted research questions, scope of the research
- Article from the web sites that are related with IoT
- Most recent journal, book, conference paper, thesis paper

Exclusion Criteria:

- Articles that are not written in English
- Unclear research scope and objectives
- Paper that are not related with software engineering, IoT, IoT development methods, IoT business models will be excluded from the study
- Incomplete and biased paper will be excluded

3.5 Quality Assessment

Quality assessment is the fundamental activity to ensure the quality of the whole literature review process. A checklist has been designed to assess the quality of this Systematic Literature review process based on the article published (Sulayman and Mendes, 2009).

Table 4: Quality Assessment Checklists

No	Questions	Answers
1	Does the article define the research scopes and objectives clearly	Yes/No/Partially
2	Does the article illustrated the Research questions clearly	Yes/No/Partially
3	Does the study followed a suitable Research method	Yes/No/Partially
4	Is the methodology used in the current research suitable to address the exact problems	Yes/No/Partially
5	Is the research studied some previous literature	Yes/No/Partially

6	Does the findings of the research satisfy the current research questions	Yes/No/Partially
7	Does the research validate its findings carefully to endure the highest quality	Yes/No/Partially

3.6 Primary Study Selection

With the search strings and the resources defined in the searching procedure stage, I have started the selection of primary studies that will be needed for this SLR.

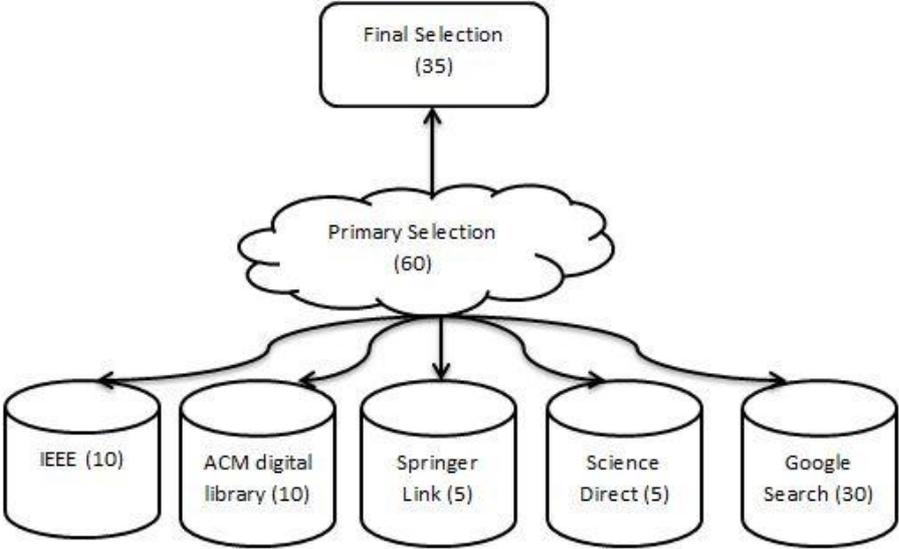


Figure 2: Selection of primary studies

Initially, I have started to look for relevant articles from the defined journals and independent search in Google using the key words defined. Approximately, I have collected 60 resources to study. Among them, I have selected 35 articles for the final reviews whereas rests of the resources were not efficient enough to satisfy the research needs. Primary selection was depend on the title and the abstract whereas final selection was done by clearly analyzing the title, abstract, introduction and the conclusion part from each of the relevant articles.

3.7 Data Extraction

Data extraction is the most important stage in Systematic literature review. The main objective of this stage is to extract the required data that will be used for the whole research to answers the research questioned defined. Data extraction will be performed to summarize the findings that will be found during the related studies. The findings from the studies will be extracted with the following information:

Table 5: Data Collection

Number	Data Item	Descriptions
DI1	Study Identifier	Study Resource Id number (e.g ID:01)
DI2	Title	Title of the Paper
DI3	Authors	Name of the Authors
DI4	Country	Country of Authors
DI5	Publication Info	Name of the Publication Place
DI6	Publication Type	Type of the study (e.g:conference/workshops/journal)
DI7	Publication Source	Link of the study materials
DI8	Abstract	Abstract of the paper (If available)
DI9	Study Aim	Aim of the papers
DI10	Study Findings related to RQ	Major findings from the study related to RQ
DI11	Key Notes	Key notes from the study to be used to answer the RQ

To collect the relevant data needed to answer all the research questions each and individual research paper will be studied carefully. The collected data will be recorded to a spreadsheet with the information needed for RQ's. The information can be a piece of textual data or can be more complex information that will be used to answer the RQ's.

To avoid the bias during the collection of data the whole data collection process is divided into two halves. In the first stage of data collection phase, the data for RQ1 will be stored where in the later stage the focus will be turned to RQ2. To validate the data that will be extracted the student himself will read and fill the data extraction form. The supervisor of the thesis will check the quality of the data extraction and provide some improvement suggestions if needed. There can be several discussions about this step between the student and the supervisor himself to ensure the best understanding about the studies and to avoid some unnecessary conflicts during the whole review process.

4 DATA SYNTHESIS & RESULTS

In data synthesis stages all the findings will be organized to show the results from the study. In some cases where needed the data were tabulated to show the findings from the study. The following sections will illustrate the results and the synthesis process for each research questions:

4.1 Research Question 1

RQ1 stated that “How the software development have changed because of IoT?”. To simplify the questions, it was divided into two sub questions.

RQ1.1 states that ” How the characteristic of software development problems is changing because of IoT?”

In relevance to the RQ1.1 several challenges have found from the SLR in IoT development. To simplify the overall challenges found from the SLR, the whole findings are categorized in five clusters where each of the clusters contains several items related to IoT challenges.

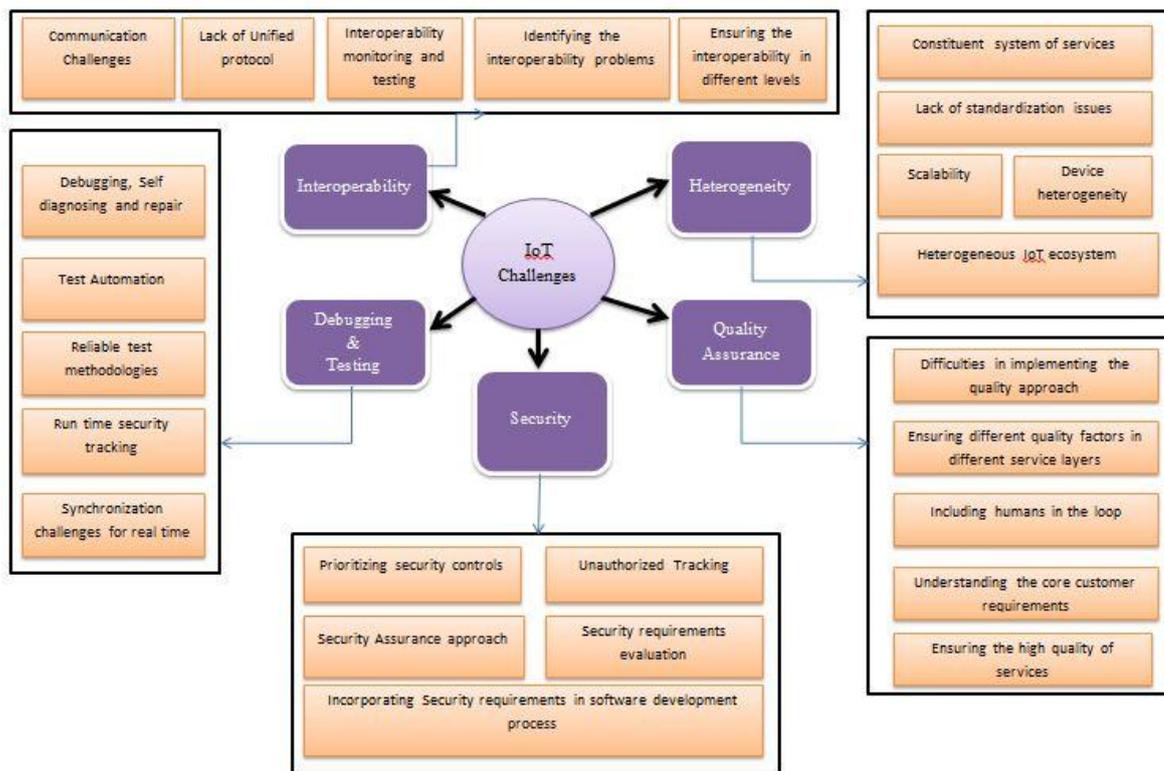


Figure 3: IoT related challenges

Figure 3 shows that the major challenges in IoT software development rely on several areas such as: security, interoperability, heterogeneity, debugging & testing, and quality assurance. Although, most of the challenges are quite known in Software development but when it comes for IoT software development the characteristics and the effects of those challenges are quite different and critical in practice to handle. Most of the studies discussed the technical aspects of IoT development challenges that can pose serious threat to the IoT software development. In this paper, the five critical challenges found from the systematic literature review will be discussed one by one in the following sections.

Security

The most common challenges in IoT development are handling different security issues that discussed by several papers. Figure 3 illustrates the issues that are responsible to cause security challenges in IoT development that have found from the SLR. IoT raises a wide range of security challenges due to the vast number of devices interacting together and if some of the IoT devices depend on open source software with vulnerabilities updating and controlling of such system would be difficult in nature (“Rajeev et al.,” 2015.).

Furthermore, the complex nature of IoT makes it vulnerable to several security issues and there are numerous challenges in incorporating security in software development. In an environment like Agile integrating security requirements as a backlog item in an iterative development might be difficult as other functional and non-functional requirements are given highest priority. Moreover, often security requirements are given less concern in the initial stage of software development. Security assurance approaches can be complex to define, plan and execute as it needs to involve the experts and other stakeholder. Additionally, due to the huge number of security controls prioritizing and controlling them would be problematic in nature in the software development. On the other hand, security objectives, requirements, control might change over time due to the diversified nature of IoT application can pose critical damage to the whole development processes (Duc et al., 2017). Security issues are important factors not only in IoT software development methods but also for other terms like as: IoT networking architecture, IoT hardware etc. and any types of vulnerabilities such as: unauthorized access on those systems can cause a serious security threat for the whole IoT system. The alarming nature of IoT security issues alter the way software developer plan and execute and it allows developers to reorganize their thinking process to handle different security challenges on IoT development.

Interoperability

In software systems interoperability means the ability to communicate and working together with different types of other software systems to achieve a common goal. The term interoperability is directly related with IoT because of the complex structure of it. IoT devices connect with other vendors, software, and network or hardware technologies to deliver the services. Each of those devices has their own standards, communication protocols, operating system which makes it difficult to guarantee that they are understanding each other and interacting correctly to exchange information smoothly. To achieve and enable the full potential experiences of different connected devices IoT ecosystems requires seamless programmability and maintainability for handling the interoperability issues (“IoT Interoperability,” 2014). On the other hand, the heterogeneous nature of IoT application creates problems with the interoperability issues where interconnection between multiple domains can be difficult to handle. As, IoT application encompasses several connected devices that may have their own communication standards

and protocol to share data and exchange information which leads to the lack of unified standardization. The presence of standardization can ensure the successful integration of IoT devices which can reduce the problems of interoperability (Al-Qaseemi et al., 2016). Complying with the standardization issues can be an optimal solution of interoperability issues in IoT development but there remains a substantial burden for the developers to understand the real interoperability problems, implement the logic and test solutions based on the unified standardization accordingly (Grace et al., 2016b). Also, the real problems arise in terms of continuously monitoring the interoperability issues and testing out them in the different levels. For example, at the device levels different software and hardware features must be interconnected to enable the interoperability, at the protocol level different communications standards such as: (ZigBee, Bluetooth, WIFI) must be configured and at the data level a flexible protocol that can translate all the data from the sensors and the other devices into a uniform format through data processing and processing functions must be defined (Suriya et al., 2017). Moreover, often the interoperability, unified standardization issues are given less priority during the development process as current approaches of IoT development are in their infancy to handle those issues during the initial stage of development life cycle.

Heterogeneity

The word heterogeneity indicates the diversity and in software development the term means consisting of different components of different characteristics within a system. Heterogeneity is the fundamental characteristics of IoT application where several devices connect together to build complex system. The characteristics and nature of the several devices varies from their types, interaction modes, as well as different operating systems. Also, the numbers of devices are increasing in IoT systems which are evolving continuously to gain better services. This complex diversity poses a critical challenge for the application developer to address all the issues throughout the development life cycle (Chauhan et al., 2016). On the other hand, in order to increase the efficiency of IoT application, service developers want to add value on top of the existing IoT system that may even integrate millions of heterogeneous devices. These systems can implement a variety of unknown networking protocols, communication standards and connectivity options. Even within one and individual IoT standards there can be multiple options to

communicate with each other (Van den Abeele et al., 2015). These problems can cause serious damages to the overall software development life cycle in terms of controlling, managing and prioritizing the different communication protocols. On the other hand, in terms of constituent system of services the current traditional systems and software engineering methods are quite inadequate to address the interoperability issues due to the inherent emergent behavior and operational and managerial independence (Maia et al., 2014). As a result, addressing the interoperation of constituent system of services with the actual physical devices is one of the key challenges that need to figure out in IoT software development. Also, the lack of unified standardization for choosing protocols, networks, managing different layers due to the heterogeneity of connected devices is a critical challenge for IoT software development that needs to be considered enough to mitigate the underlying barriers in IoT development (Al-Qaseemi et al., 2016).

Debugging and Testing

The extremely dynamic nature of IoT application poses additional challenges. For example, debugging and testing of IoT systems can be challenging due to the huge number of devices, heterogeneity, unreliable connectivity and sometime invisibility that may occur. Individual IoT devices with limited data collection can be much easier to debug and test but complexity arises when hundred or even thousands of heterogeneous devices deployed in a complex environment (Taivalasaari and Mikkonen, 2017). Continuous delivery has becoming the popular practices for software development in the industry throughout the years. Each level of continuous delivery it includes the automation of every phase to increase the release cycle and bring production like environment to every testing phases. Due to the diversified nature of IoT application as well as the complex physical environment automation of testing would be challenging to implement in the software development process (Farahmandpour et al., 2017). In the fast changing atmosphere of IoT design and development, reliable testing methodologies are increasingly important phase in the software development. As, developers continuously integrate newer form of devices such as: sensors, RFID and platforms, understanding the core customer requirements and evaluate them carefully throughout the initial stage of development is a significant but problematic step to implement the strategic testing design process of IoT development (Armstrong, 2017). On the other hand, IoT application works with data that need to be

synchronized in real time manner to ensure the best services is a considerable issues for IoT software development to handle.

Quality Assurance

Quality assurance is another terms that are indispensable part in software development. To increase the market value of the product it is essential to assure the quality of the product. Quality assurance is a term that systematically checks if the product and services are meeting the specified requirements and standards (Marwah et al., 2015). With the growing number of heterogeneous IoT devices the challenges with ensuring the quality of IoT application is demanding. To ensure the high quality of IoT application there are numerous things to consider such as: different communication protocols, hardware-software complexity, cross domain interoperability, energy efficiency, sensing data quality and etc (Townsend, 2016). While considering those diversified issues the real challenges come in implementing the quality assurance approaches in IoT development methods. The real challenges for quality assurance team is that on which aspects they should focus on, and how to add process predictability to the practices they are following while overcoming the associated connectivity and reliability challenges (Software, 2016). The vast number of considering issues in IoT application makes it difficult to implement the exact quality assurance approach in software development life cycle. Besides, the automation features is in the rising manner both in IoT application and the software development approaches. Automation of quality along with the exact testing strategies is a burden for the application developer in IoT development that needs to be considered. Also, the growing number of physical devices in IoT application brings challenges to ensure the different quality factors such as functional stability, performance efficiency, compatibility, usability, maintainability etc in different ranges of layers: physical layers, deployment layers, link layers, and network, application and middleware layers (White et al., 2017).

RQ1.2 states that “What implications identified IoT-related characteristics pose on software development methodologies?”

IoT application is combination of multiple complex devices from different sources that interact together to exchange data and information. The structure of IoT changes the way of software developer’s thinking process. The characteristics of IoT application bring huge

changes in software development methodologies.

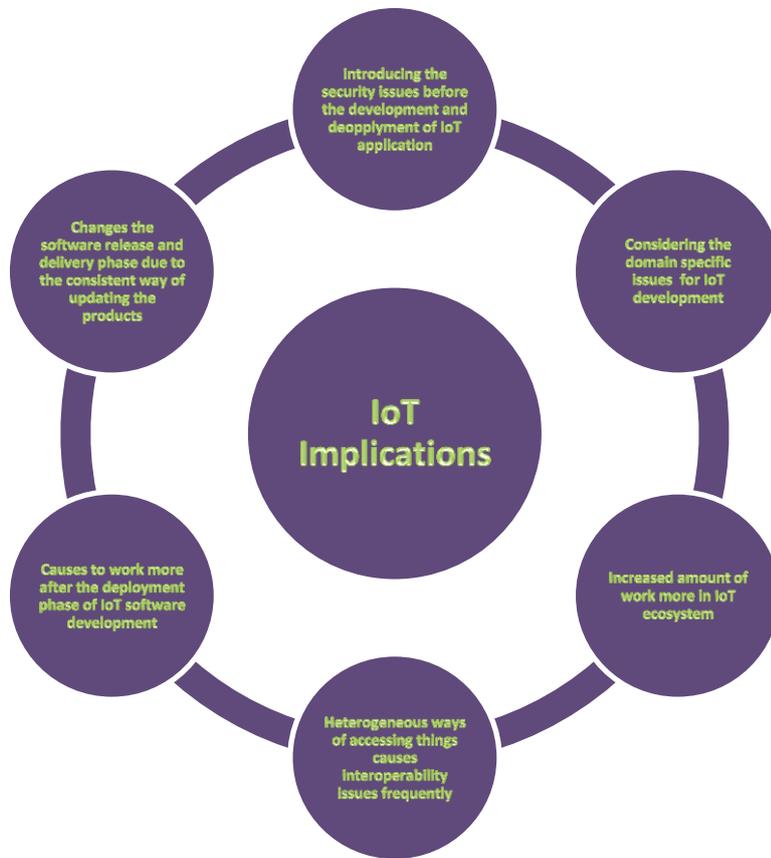


Figure 4: IoT related implications

As discussed earlier that security is one of the critical threat and major concerning issues in IoT development, the impact of security in IoT software development methodology is tremendous that organizations need to redesign the development processes. Most of the studies suggested that security issues can be addressed if challenges related to security can be introduced before the development and deployment of IoT application (Javed et al., 2017). Adding the security features after the production are less likely to be effective in terms of cost and implementation that can be tailored by adding the security features from the initial stage of design consideration phase in development processes. The idea can be an optimal solution to handle different IoT related security challenges but there remain substantial legal issues as IoT applications are combination of different hardware, software, networking protocols from different vendors. The control environment needs to be supported by a governance framework, appropriate policies and procedures,

continuous monitoring and maturity model to ensure that overall system is protected from all types of known and unknown issues. The most popular fast paced development approaches like agile, DevOps, continuous delivery, lean methodology that are being used mostly in software development but in terms of IoT development there need to consider some domain specific issues. For example, considering security issues as a backlog item in an iterative development, security requirements evolution, managing the 3rd party components, prioritizing the different security challenges are tailoring the way companies are using those fast paced development approaches. Therefore, there are substantial changes happened in the software development process due to the complex nature of IoT security issues.

The biggest changes happened in the steps of software release and delivery for IoT. It is prominent that IoT has totally changes the way products are delivered to the public. The most critical change is the need for updating systems for the devices preferably without the user intervention. The typical scenario with updating the packages for the devices such as iPhone at the moment is that user decides whether they are going to update the version or not. These scenarios will changes in case of IoT application. IoT based application need more consistent and proper way of updating the systems as security, safety and other legal issues are directly related. Moreover, updating with one specific feature can cause damages to the other systems that are related closely with each other and even there can be thousands of developments each and every day which will pose serious challenges to handle the total software development processes (Labs, 2015).

Another issues are found during the SLR is that traditionally software development tends to happen in silos where individuals or specific teams were responsible to perform their jobs independently. This thing is not going to work for IoT development as several vendors working together needs to co-operate to work more in IoT ecosystem (Wallgren, 2017). As the demands for IoT continue to rise and consumers' expectations are increasing, it is more and more imperative for the organizations to operate as efficiently as possible. The trending nature of IoT ecosystem will change the way of software development process where integrating and managing different stakeholders can pose serious challenges.

The most exciting IoT platforms offers heterogeneous ways of access things and their data which cause interoperability problems when developers aim to create overarching, cross-platforms and cross-domain applications and eventually it eludes the appearance of vibrant

IoT ecosystems. To address the interoperability issues, the BIG IoT (Bridging the Interoperability Gap of the IoT) projects launched to enable the cross-platform, cross-domain and cross-standard IoT application and services. The BIG IoT projects connects the things, service providers and users together to build a well-organized IoT ecosystem. To reduce the interoperability gap in IoT ecosystem The BIG IoT project mandates the use of common information models. This common information models providers (platform or services) to represent the resources in a machine readable format so that (consumers or application) from a different domain, region can understand and process them. For instance, schema.org vocabularies are the shared common understandings for the search engines and other billions of web pages (Bröring et al., 2017).

As the consumer expectations are changing frequently IoT software development will have bigger impact on the software development life cycle. Traditional software development tends to be completed once the deployment steps are done whereas in terms of IoT software development deployment is not the end. With the growing expectations of post deployment content and updating features organization of IoT software development need to adjust the change and management solutions, testing over a greater development life cycle by ensuring the better user expectations.

4.2 Research Question 2

RQ2 states that “What IoT development methods have been proposed?”

In relevance to this question there are few development methods found from the SLR that directly focused on IoT development. The two broadly used IoT methodologies are Ignite|IoT development methodology and The IoT Methodology. The following sections will illustrates these two methodologies:

Ignite|IoT development methodology

Ignite IoT methodology is becoming increasingly popular for the companies who wants to transform towards IoT based business. It has two main parts such as: Ignite|IoT Strategy Execution and Ignite|IoT Solution Delivery (Slama Dirk et al., 2015). Each of the main parts has their own elements that are being followed to implement IoT based business models. The following sections will discuss about the steps of Ignite|IoT development methodology.

Table 6: Ignite|IoT Development Methodology

Ignite IoT Development Methodology					
Ignite IoT Strategy Execution			Ignite IoT Solution Delivery		
IoT Strategy	IoT Opportunity Identification	IoT Opportunity Management	IoT Solution Life Cycle	IoT Building Blocks	IoT Project DB
Initiation	IoT Center of Excellence	IoT Platform			

Ignite|IoT Strategy Execution

This phase helps organizations to understand the strategy for IoT based businesses and goals. It has six main areas: IoT strategy, IoT opportunity identification, IoT opportunity management, Initiation, IoT center of excellence and IoT platform.

IoT strategy is an important terms that the companies need to understand properly. It defines the organizations goals, objectives and the detailed description on how the strategic alliances and partner ecosystem will be developed accordingly. In this part of Ignite|IoT strategy execution organizations defines their thinking process about the IoT business, budget planning as well as the management of IoT roadmap.

IoT opportunity identification is more about the generation of innovative ideas that will be elaborated through a structured manner. The employees, customers and the developers will take part in the idea generation process where ideas will be derived from a given context, such as the companies’ value chain, For example.

IoT opportunity management is the following part of the earlier stage where the most prominent ideas selected from the IoT opportunity identification step. In this part, the most suitable and promising ideas will be refined to prepare a detailed business models that will help to assess the feasibility and the business case.

Initiation is more about initializing of the project after making the decision about the opportunity approved by the organizations. In this initiation stage, management has to

make decision about the setup of project initiation step: as a dedicated internal project, as a spin-off, or as an M&A project.

IoT center of excellence can foster IoT application development by delivering the IoT consulting and change management supporting. For example, an IoT maturity assessment testing can be done that will help an organization to understand better whether it currently stands with respect to IoT adoption or not.

IoT platform is the stage where organizations define their application platforms, connectivity solutions, technical and functional standards. Sometimes, it makes sense for the organizations to establish a shared development platform that can be used by multiple projects to develop their solutions.

Ignite|IoT Solution Delivery

Ignite|IoT solution delivery is key part of Ignite|IoT methodology that incorporates different disciplines together within a single project to plan, build and run an IoT solution. It can be broken down in several steps such as: IoT solution life cycle, IoT building blocks and IoT project DB.

IoT Solution Life Cycle focuses on planning, building and running an IoT solutions that is divided in two categories such as: Initial Project Design and Project Workstreams.

Initial Project Design is a blueprint that builds on the elements defined as part of the generic IoT Building Blocks, including project self-assessment using IoT Project Dimensions, solution architecture using IoT Architecture Blueprints, and technology selection using IoT Technology Profiles. On the other hand, Project work streams define the top-level work streams typically found in an IoT solution project. A checklist for each work stream is provided, along with a list of common dependencies between the work streams.

IoT Building Blocks contains three elements: Project Dimensions, Architecture Blueprints and Technology Profiles.

Project Dimensions are used for self-assessment, project comparisons, architecture and technology selection etc. that can be used as a precursor for the formal requirements. Architecture blueprints add new architectural perspectives for IoT projects and provide a structure for integrating various architectural perspectives that will be required. Technology Profiles usually describes and identifies the most suitable technology that are required into

the overall IoT architecture.

IoT Project DB provides the references for the whole IoT projects

The IoT Methodology

The IoT methodology is an iterative process inspired by lean startup and design thinking approach. The aim of this model is to deliver a loosely structured ecosystem model of mutual value for all who will take participate, driven by sharing, collaboration, community and learning. It's an iterative methodology that developed iteratively. The iteration process includes several steps such as: CoCreate, Ideate, Q&A(Questions and Answer), IoT OSI, Prototype and Deploy (Tom, 2014).

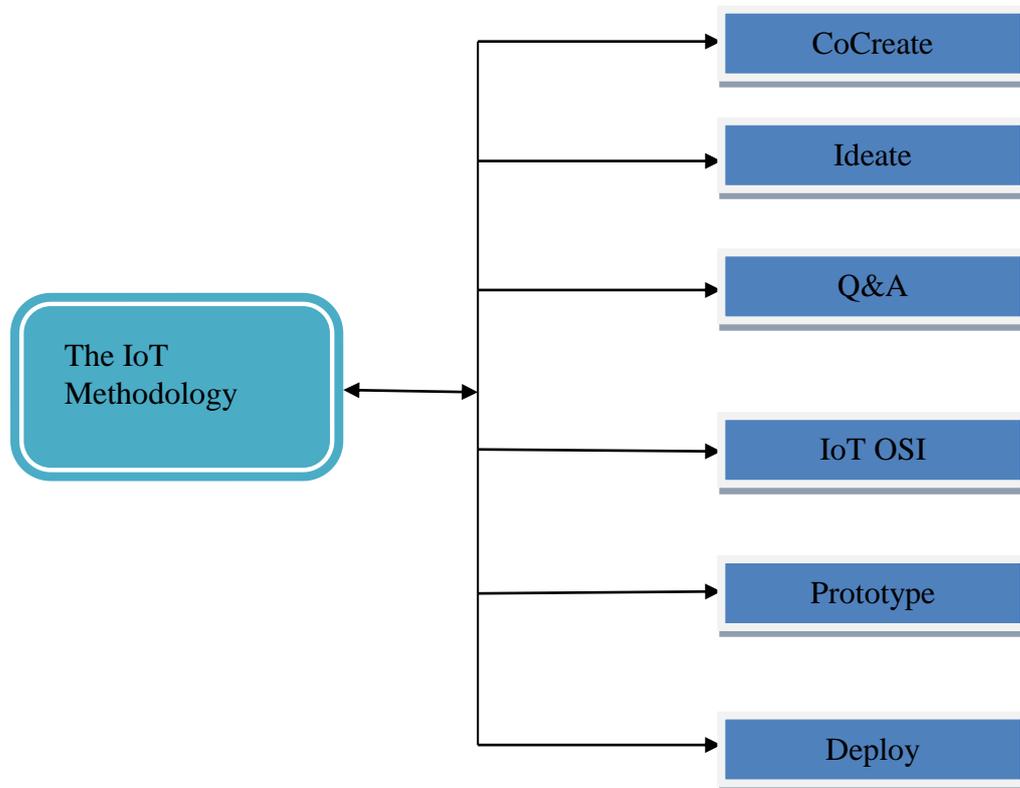


Figure 5: The IoT Methodology

CoCreate

The main purpose of this initial step is to arrange a workshop for communicating and interacting with the stakeholders to identify the pain problems. The problem identification is done through a non-technical way as there can be different types of stakeholders. The possible task that will be accomplished through this step can be summarized in the

following ways:

- Discuss the actual domain of the problems and narrow down the biggest pain problems.
- Consider influences and outcomes of these problems to contextualize with the physical environment.
- Identify the different knowledge areas contributing the solutions.
- User interface mockup creation to allow users to visualize and understand the possible solutions.

Ideate

This step is more about interacting with the team who will take part in the development of the application to simplify discussions to communicate requirements with designers, implementers and project managers. In this step the IoT canvas is used to simplify and summarize the findings from the idea generations. IoT canvas is an adaptation of the Business Model/Lean Canvas, it assists validation and identification of MVP (Minimum Viable Product) requirements for projects.

The purpose of this step can be summarized in the following ways:

- Providing a common reference for brainstorming and discussing IoT ideas, use cases and projects
- Quick and easy for the prospects and practitioners.
- Standardized terminology for designer & developer centric approaches.
- Opens discussions between different stakeholders.
- Identify things in the physical environment including sensors, actuators, output and controllers that make up of endpoints interacting with things.
- Define data models and middleware requirements for endpoints to connect IoT services.
- Summarize the third party web services and define the key actors
- Sketch user interface widgets.

Q&A

The questions and answers step is more about brainstorm and analyze to transform the soft requirements into hard requirements. A series of survey can be conducted with early user groups to discover the best of conceptualizing a project and choosing the right most appropriate solutions for IoT OSI layers to assist project development. The purpose of this step can be summarized in the following ways:

- Closed the gap between idea and implementation by providing a business value or project outline.
- Identifying the feasibility of the solutions, defined problems, skills, practicality by validating the scope of the idea.
- Problem domain analysis by checking up the solution summary.
- Solution topology design and infrastructure considerations.

IoT OSI

Requirements map to valid architectures, infrastructure and business frameworks. Use an IoT OSI model that contains different components to define the IoT OSI architecture models. The components of IoT OSI are defined below:

End Points have a number of roles including producing data, receiving commands, providing services for management, discovery, persistence and local logic. For instance, Temperature sensors, LED, RFID reader etc.

Connectivity the wire level communication protocols used to connect the endpoints with the middleware. Examples: TCP/IP, IPv6, REST, ZigBee etc.

Middleware Standardized layers using connectivity drivers, to translate and connect disparate sources. Example: Dowse, OpenHub, The thing system, WebSphere etc.

IoT services Typically provides the smarts for the IoT and power where processing on endpoints or middleware is not sufficient. Examples: persistence, automation, intelligence, third party integrators etc.

Apps User facing front end applications that facilitate using IoT services, in turn can be used to manage and automate endpoints, and generally orchestrate an internet of things. Examples: WidgIoTs, SmartLiving Web&Mobile Apps, Freeboard etc.

The purposes of this step are given below:

- Breaks down and the simplifies the components of the IoT into an “Internet as an operating system” paradigm.
- Makes it easy to digest the scale of internet of things and all of it’s intricate components.
- Define scope areas for domain specialists to focus efforts.
- Simplifies communications with end users, stakeholders and development teams.

Prototype

Standardized toolkit assist building prototype which iterates towards minimum viable product. The purpose of this step is explained in the following ways:

- Define an ethos with Lean startup mentality, designing prototype to test, measure and iterate.
- Uses standard building blocks to build IoT products fast.
- Allows developers and practitioners to focus on to what they are good at, and using the IoT toolbox for all the layers.
- Use technologies and tools conducive for rapid prototyping.
- No major concern for authorization, scalability or UX in early iterations.

Deploy

Continuous deployment assist closing the feedback loop, enhances knowledge sharing and collaboration opportunities. The purpose of this step can be summarized in the following manner:

- Package up IoT projects to share with the world, defining a standard IoT project structure
- Cuts deployment time and integrate IoT Unit Testing.
- Collates code for a whole project with all layers of the IoT OSI, along with all configurations.
- Makes it easy to share, collaborate and build on top of existing solutions.
- Enhances knowledge sharing.

RQ2.1 states that “How the proposed methodologies are dealing with current challenges of IoT development?”

Apart from the two broadly used IoT methodologies: Ignitie IoT and The IoT Methodology, **DevOps** is one of the best industry practices in software development that first coined by Patrik Debois in 2009. New Relic explained that DevOps is the mindset that simply combined the software development and operations together in order to emphasize the communication, collaboration and cohesion between the several scattered developers and IT operations team members (“What is DevOps?,”).

There is no predefined Phases of DevOps rather organizations tend to design their DevOps platform depending on their needs and requirements. The typical DevOps Phases includes the following steps (Virmani, 2015):

- **Continuous Planning** DevOps starts with a proper plan if needed based on the small portions of customer requirements to adjust, get feedback and react on feedback.
- **Continuous Integration** A continuous process that refers to integrate early in the development team to validate how the changing of code behaves continuously.
- **Continuous Deployment** It is known as the hearts of DevOps operations where automation is the key features to maintain continuous delivery of the software.
- **Continuous Testing** Automate the testing of each delivery into the software before merging to the production environment.
- **Continuous Monitoring** To monitor the performance of the application and adjust any surprises in timely manner.

Identified challenges handled by DevOps:

Security:

In terms of dealing with security issues DevOps approaches have bunch of technologies such as: Veracode, EMMA and Inspec can be used to run security analysis and identify the vulnerabilities and other compliances issues. There are some other tools for unit testing such as: Junit, Xunit, Mstest are available on the technology stack. Regarding the code quality analysis tools SonarQube, Coverity, Cobertura are being widely used for code quality analysis in IoT application development. Also, various security standards for device and user authentication identity such as: SAS-based token per unique publisher,

certificates, TLS authentication support are available during device development and security testing. Often, the security testing is being carried out during the development phases where continuous integrations are inevitable (L. Subbarayalu, 2017a). Although DevOps approaches have bunch of technologies and tools to handle different security issues but again there are challenges that the organizations needs to deal with. Teams need to be educated and aware of different security threats. For example, running services with least privileges during security design and development, using tamper-resistant protocols, authorizations policies and filtering can be difficult in practice. As discussed earlier, security testing as part of continuous integration is common practice in DevOps manner which allow developers to consider several issues like: integrate the appropriate security tools, customize different security compliances, generate reports on fix and non-fixing issues early in the development phases to minimize risks can be a tremendous pressure due to the vast amount of end point devices connected to the IoT application. On the other hand, follow the security guidelines for large amount of devices, software and hardware and tracing each of the recordings can be huge challenges to deal with using DevOps approaches (L. Subbarayalu, 2017b).

Interoperability & Heterogeneity

Interoperability and Heterogeneity are the key challenges that need enough consideration while developing an IoT application. As the devices connecting together are in the rising manner, the challenges of both this terms are also considerable issues in DevOps development approaches. In terms of traditional software development DevOps can be a feasible solution but when millions of devices and networking protocols work together that can be a real challenge to handle using DevOps approach. For example, in it's each steps like CI, CD DevOps implements the automation features which usually add numerous types of tools to connect together and on top of it if IoT application comes where thousands of connecting points will be added automation, monitoring and managing would be time consuming as well as costly for the organization. On other hand, to provide a scalable business organization needs a scalable infrastructure where millions of dot points connecting together can send and receive data. To provide this features, DevOps offers cloud environment where a scalable infrastructure can be implement but again the challenges lies with the skilled labor, effort and cost as different vendors can have different

opinions ((Lavanya Subbarayalu, 2017). Interoperability and Heterogeneity issues arise when millions of dumb devices work together and with rapid growth of automation in DevOps approaches poses a challenge for the organization that needs to deal with more.

Debugging, Testing & Quality Assurance

In a connected world where competition is increasing organizations need to ensure that the debugging, testing and quality assurance issues are handled properly to re-confirm the best services of their IoT application. In a real complex IoT structure automation in testing is challenging and fortunately DevOps engineers found some comprehensive tools that can be handy in practice for IoT application testing. For example, Subbarayalu explained that Selenium, Jmeter, security testing tools such as: Veracose, EMMA, CA Lisa, Inspec, Robot Framework and Performance tools can be used to implement the fully featured automation testing in Phases like CI and CD (2017a) . On the other hand, although there are a bunch of testing tools available to automate the feature once again the challenges come for Big complex IoT application. Usually, IoT application needs end-to-end testing of multiple devices interconnected where controlling most of the devices development from other organizations would be difficult except the own device development. In that case, a robust test cases, simulated test devices and test environments like cloud are necessary elements to test IoT application to numerous operating systems and devices (Sand, 2015).

Quality assurance is kind of a burden for IoT application developers. Also, it cannot be denied that QA is one of the key terms that organizations need to ensure for better services of their application. In a typical DevOps environment quality integrates with the testing and development process and takes a collaborative approach. Quality is ensured in testing and delivery life cycles where both the testing and development teams take part in it from the entry level of the development life cycle. In terms of code development, the QA team pushes the new code into the production environment to facilitate the seamless deployment of infrastructures and tools. Therefore, it ensures the continuous release of code in a consistent manner and thus it just doesn't remove the bugs from the code but it is built on the overall application process (Lopez, 2017). This process can be efficient for traditional systems and services but when the terms come with IoT application challenges arise implementing this process due to the vast amount of layers, services, different vendors etc. On the other hand, the unpredictability with IoT application allows continuous changes and updates which

cause the harm for IoT application development. According to Sand, In a DevOps environment often the Continuous delivery service model works as an optimal solution for these problems where the updates and changes are directly pushed from back-end through continuous delivery life cycles (2015). But, this architecture doesn't hold for the IoT application where updates and changes might depends on several other devices, networking protocols and standards.

Ignite IoT methodology is designed based on several practices that composed in two categories (Slama Dirk et al., 2015):

Ignite | IoT Strategy Execution: This phase helps organization to transform towards the IoT-based business models by defining the IoT strategy and prepare the organization for IoT adoption. This perspectives helps the management, CEO,CTO of the company to define the business goals and understand the potentiality of IoT business. IoT strategy, IoT opportunity identification, IoT opportunity management and project initiation etc are the practices that will be defined and explained during the Ignite | IoT strategy execution steps.

Ignite | IoT Solution Delivery: Where the Ignite | IoT strategy execution focuses on the management perspective, the Ignite | IoT solution delivery will focus on the individual project to deliver the solution. Usually, this phase defines the interfaces to the related assets and the organization. It supports the product managers and project managers in the planning and execution of IoT projects where plan, build and run works as a IoT solution life cycles, IoT building blocks that contains the reusable artifacts from the successful projects in the form of IoT project dimensions, IoT architecture Blueprints and IoT technology profiles.

Identified challenges handled by Ignite:

Security has been considered an important element in Ignite IoT methodology by giving priority in the IoT solution design phase. Once the solution design has been completed, the final tasks in the project initiation phase is to set up the organizational structures where different IoT project work streams, project management issues are discussed and handled and security is one the issues that has been undertaken during considering about the different cross-cutting work streams that may have impact on the projects. Ignite IoT defines a strategy for ensuing and validating the security that requires policies and process including the terms such as: risk assessment, security architecture, security

implementation, security testing and validation. Risk assessment helps to identify the highest priority threats and the profiles of the most likely attackers, security architecture is designed to show how different security mechanisms will be integrated into the solution architecture, security implementation included the detailed plan for implementing the security architecture for cross cutting issues that involves all other work streams, security testing help to identify the potential flaws in the security architecture and implementation. On the other hand, Ignite IoT methodology uses waterfall or agile way of plan, build and run solution life cycles depending the different projects that may have own way of handling security threats. Ignite IoT methodology has a well-organized way of handling security issues but to implement and execute different security testing strategy may need some outside experts that can be costly and some other legal issues have to defined that can result in project delays or time consuming issues as well.

Interoperability & Heterogeneity issues in Ignite | IoT methodology have been undertaken from the asset life cycle management steps as these challenges are really common in IoT application. A dedicated role is being created with responsibility to look for Interoperability and Heterogeneity issues from an asset life cycle management perspectives across all the work streams. For example, asset activation is an important part of the life cycle usually has an impact on the devices, software, communication services etc. Communication services are one of the elements of Cross-Cutting tasks that ensure the different protocols and standards within the devices. To enable the best possible communication services, several tasks are being undertaken from the initial solution design phase. The tasks includes, collecting detailed requirements on services, including expected traffic profiles and backend connectivity and locations; Analyzing different hardware capabilities for smooth way of communication services, interaction with mobile networks and roaming ; determining the different regulatory situations in the launch market for communications protocols and standards: monitoring continuous communication behavior and impact on networks. Also, a test environment will be implemented during the build phase to check the integrated communication services provider's infrastructures. A portal will be used in the run time to monitor and manage the communication networks, status, errors etc. In terms of integrating different middleware platforms, a pre installation and configuration have been performed and analyzed during the analysis phase in order to

determine the best way of mounting the solution on asset hardware components such as: actuator, sensors gateways etc. Asset Integration Architecture known as AIA in Ignite IoT methodology has been designed in the IoT solution delivery phase under the building blocks step is an important term that defines the cross domain, cross culture and cross integration issues in different levels. For example, on the device level integration can happen through any kind of wireless communication standards such as: Bluetooth or Gateways can be interact with each other directly by Car2Car type of integration scenario. But, the fact is that to enable the different subnet of things and their integration it is important to understand the different options in the Asset Integration Architecture. Any kinds of mistakes when designing the architectures can cause serious dilemma in the whole application.

Debugging, Testing & Quality Assurance

Debugging and Testing is being performed under the cross cutting tasks in Ignite IoT methodology. A person or a team with couple of members depending on the project size is assigned to perform this cross cutting task known as solution integration and testing. The team will be responsible to implement the different release milestones, interface versions, staging mechanisms, process and end-to-end test plans, etc. to create a feasible test environment by following the plan, build and run concept. To handle the challenges with large IoT implementation testing where thousands of devices have to be tested IoT ignite proposed a pragmatic mechanism. An initial integration test with a limited number of devices can be performed by using a load generator to simulate a system with thousands of devices and to test the scalability of the back end systems. Similarly, to test the devised communication services test team required to work closely with different service providers to define a joint test plan that can be used throughout the projects. All the testing task is done through a centralize approach as tests usually have an impact to all other work stations. On the hand, use of IIC (Industrial Internet Consortium) Testbeds, a platform where application can be deployed and tested in an environment that resembles real-world conditions, makes the Ignite IoT methodology more effective in terms of Debugging & Testing. Although, the testing of IoT application by Ignite provides consistent effective approaches, in terms of ensuring the quality approach there was no specific pre-defined automatic approaches have been found. But, the fact is that IoT ignite is well-organized in

its each steps and only moves to next steps when the earlier one meets with all the criteria defined. For example, IoT strategy execution phase ensures that the company is well understood with the goals and objectives of the project as well as the customer needs. Also, organization can choose make or buy decision about the specific artifacts that will be used based on the solution design stage. On the other hand, IoT uses the waterfall or agile approaches of project management depending on the diversity projects. Both the waterfall and agile are well structured with the quality management issues that make Ignite IoT methodology a handy approaches for the organization to develop IoT application.

RQ2.2 states that “What are the limitations, similarities and differences between those methodologies?”

Table 7: Comparisons of IoT development methodologies

Ignite IoT Development Methodology	The IoT Methodology
Ignite IoT development methodology is monolithic approaches that reuses the existing practices like: Waterfall or Agile depending on the types of the projects,	The IoT methodology is also a monolithic approach that was inspired by Lean startup and design thinking process.
Ignite IoT development methodology is based on the IoT strategy management and project execution that consists two main parts: Ignite IoT strategy execution and Ignite IoT solution delivery.	The IoT methodology is iterative and consists of six iterations steps: co-create, ideate, Q&A, IoT OSI, Prototype and Deploy.
Ignite IoT development methodology has two domain specific practices such as: Project Dimensions known as PD and Asset Integration Architecture known as AIA	The IoT methodology came up with the two special practices: IoT canvas and IoT OSI reference model that make it suitable for IoT project.
Ignite IoT solution delivery supports the plan, build and run approached in IoT solutions delivery approaches	The IoT methodology follows the continuous deployment after implementing a minimum viable product through a prototype.

It is very prominent that both these methods are well organized and highly monolithic

method that reuses some existing practices. For example, Ignite|IoT development methodology required to use whether Waterfall or Agile way of development methodology depending on the project sizes whereas the IoT methodology inspired by Lean startup and design thinking approaches (Jacobson et al., 2017). Ignite|IoT methodology comes with two domain specific practice such as: Project Dimensions (PD) and Asset Integration Architecture (AIA). The purpose of using project dimension practice in Ignite|IoT development methodology is to conduct projects self-assessment, comparing different IoT based solutions and selecting the suitable solution architectures and technologies to be used in the project. The Asset Integration Architecture is used to define the gateways, devices, and services and their functionalities within the projects which is so domain specific in IoT solution. On the other hand, what is so special in The IoT methodology is the use of IoT canvas and IoT OSI reference models. The IoT Canvas is an adoption of business model is used in the brainstorming sessions to validate the minimum viable products in lean startup and design thinking processes. The IoT OSI reference models is used to define what to do, what to change and when to complete, and provides guidelines to apply the practices effectively. The use of IoT OSI reference models makes the practices easy to understand, use and change accordingly when needed. Ignite|IoT development methodology comes with the process of decision making before starting developing the actual product in the Ignite|IoT strategy execution phases whereas The IoT methodology is an iterative process with several practices that supports the decision making approaches. Both the Ignite|IoT development methodology and The IoT methodology are refined with the existing practices and sources but do not share the common ground needed to allow them readily composed into an effective methods. Also, implementing of both this methodologies requires enough human resources as well as common ground for involving the third party resources can bring numerous challenges for the organizations to handle.

5 DISCUSSIONS AND CONCLUSIONS

Systematic literature review is one of the difficult tasks that need to be done by following the step by step guidelines defined in the review protocol. The current research study is done by following the review protocol defined and implementing the protocol is one of the toughest and time consuming tasks as well. There was couple of iterations before starting and executing the review guidelines. The overall approaches were planned and well-structured that satisfied the research needs. The primary objective to carry out a systematic literature review was to investigate different IoT related challenges and implications in software development. Also, to analyze more how the current approaches of IoT software development methods are dealing with the identified challenges and implications.

In this form of systematic literature review, I have investigated the current challenges and implications of IoT software development as well the current approaches of IoT software development methods. The overall search process for this SLR was comprehensive followed by the inclusion and exclusion criteria. The data collection and the synthesis steps were done as the guidelines defined. The numbers of main research questions investigated in this SLR were two and further divided into four sub questions.

Based on the study above it is supported that there are several IoT related challenges and implications are still creating difficulties in software development methodologies. Challenges like security, interoperability, heterogeneity, debugging & testing, quality assurances are altering the traditional way of software development methods to think about more broadly and widely. These IoT related challenges are allowing software developers to redesign their thinking process by working and collaborating more IoT ecosystem, involving the security requirements from the very beginning of software development process, changing the software release and delivery life-cycles etc.

There have been several industry best practices found during the SLR that are being used commonly. Practices like DevOps, Ignitie IoT Methodology, The IoT methodology are known as the most commonly used approaches for IoT development. These methodologies have their potentialities and limitations as well. There are certain challenges and implications that these methodologies are limited in scope to handle. Also, these methodologies are monolithic in approaches that combines different other approaches like

agile, lean etc. These monolithic approaches can bring other challenges for the industry where implementing and executing the practices can be error prone and time consuming as well. To conclude, based on the study from the SLR it is prominent that the current challenges and implications of IoT software development can play a vital role to result in the project failures where the existing industry best practices need to be applied more broadly and widely to handle those issues. Also, there are more opportunities to do more research study on this area that can help to redesign the shape of IoT development world.

REFERENCES

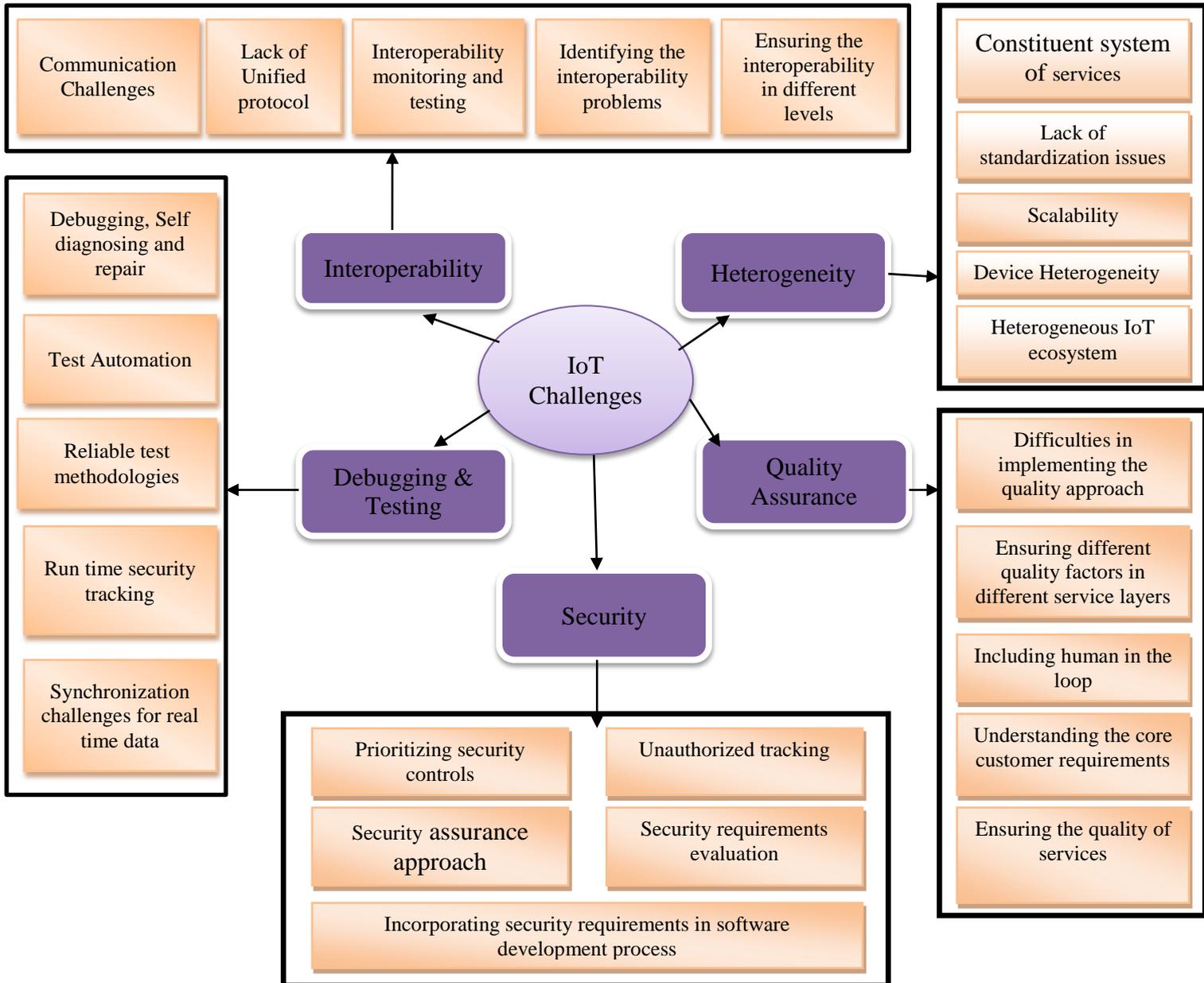
- 10 Real World Applications of Internet of Things (IoT) - Explained in Videos, 2016. . Anal. Vidhya.
- Rajeev Alur, Emery Berger, Ann W. Drobni, Limor Fix, Kevin Fu, Gregory D. Hager, Daniel Lopresti, Klara Nahrstedt, Elizabeth Mynatt, Shwetak Patel, Jennifer Rexford, John A. Stankovic, and Benjamin Zorn September 22, 2015. Systems Computing Challenges in the Internet of Things. white paper prepared for the Computing Community Consortium committee of the Computing Research Association. <https://cra.org/ccc/resources/ccc-led-whitepapers/>
- Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. EBSE Technical Report EBSE-2007-01.
- Al-Qaseemi, S.A., Almulhim, H.A., Almulhim, M.F., Chaudhry, S.R., 2016. IoT architecture challenges and issues: Lack of standardization, in: 2016 Future Technologies Conference (FTC). Presented at the 2016 Future Technologies Conference (FTC), pp. 731–738. <https://doi.org/10.1109/FTC.2016.7821686>
- Armstrong, C., 2017. Debug and analysis considerations for optimizing power in your internet of things design, in: 2017 IEEE International Symposium on Electromagnetic Compatibility Signal/Power Integrity (EMCSI). Presented at the 2017 IEEE International Symposium on Electromagnetic Compatibility Signal/Power Integrity (EMCSI), pp. 15–19. <https://doi.org/10.1109/ISEMC.2017.8077983>
- Bröring, A., Schmid, S., Schindhelm, C.-K., Khelil, A., 2017. Enabling IoT Ecosystems through Platform Interoperability [WWW Document]. InfoQ. URL <https://www.infoq.com/articles/enabling-iot-platform-interoperability> (accessed 1.5.18).
- Chauhan, S., Patel, P., Sureka, A., Delicato, F.C., Chaudhary, S., 2016. IoT Suite: A Framework to Design, Implement, and Deploy IoT Applications: Demonstration Abstract, in: Proceedings of the 15th International Conference on Information Processing in Sensor Networks, IPSN '16. IEEE Press, Piscataway, NJ, USA, p. 37:1–37:2.
- Duc, A.N., Jabangwe, R., Paul, P., Abrahamsson, P., 2017. Security Challenges in IoT Development: A Software Engineering Perspective, in: Proceedings of the XP2017 Scientific Workshops, XP '17. ACM, New York, NY, USA, p. 11:1–11:5. <https://doi.org/10.1145/3120459.3120471>
- Farahmandpour, Z., Versteeg, S., Han, J., Kameswaran, A., 2017. Service Virtualisation of Internet-of-Things Devices: Techniques and Challenges, in: 2017 IEEE/ACM 3rd International Workshop on Rapid Continuous Software Engineering (RCoSE). Presented at the 2017 IEEE/ACM 3rd International Workshop on Rapid Continuous Software Engineering (RCoSE), pp. 32–35. <https://doi.org/10.1109/RCoSE.2017.4>
- Favaro, J., 2015. Strategic research challenges in the Internet of Things, in: 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS). Presented at the 2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS), pp. 1–1. <https://doi.org/10.1109/RCIS.2015.7128856>

- Glass, R.L., 1994. An assessment of systems and software engineering scholars and institutions. *J. Syst. Softw.* 27, 63–67. [https://doi.org/10.1016/0164-1212\(94\)90115-5](https://doi.org/10.1016/0164-1212(94)90115-5)
- Grace, P., Pickering, B., Surridge, M., 2016a. Model-driven interoperability: engineering heterogeneous IoT systems. *Ann. Telecommun.* 71, 141–150. <https://doi.org/10.1007/s12243-015-0487-2>
- Grace, P., Pickering, B., Surridge, M., 2016b. Model-driven interoperability: engineering heterogeneous IoT systems. *Ann. Telecommun.* 71, 141–150. <https://doi.org/10.1007/s12243-015-0487-2>
- History of IoT | Background Information and Timeline of the Trending Topic [WWW Document], 2016. URL <https://www.postscapes.com/internet-of-things-history/> (accessed 10.7.17).
- Hua, K.A., 2016a. Internet of Things: Challenges and Opportunities for Collaborative Technologies, in: 2016 International Conference on Collaboration Technologies and Systems (CTS). Presented at the 2016 International Conference on Collaboration Technologies and Systems (CTS), pp. 613–614. <https://doi.org/10.1109/CTS.2016.0114>
- Hua, K.A., 2016b. Internet of Things: Challenges and Opportunities for Collaborative Technologies, in: 2016 International Conference on Collaboration Technologies and Systems (CTS). Presented at the 2016 International Conference on Collaboration Technologies and Systems (CTS), pp. 613–614. <https://doi.org/10.1109/CTS.2016.0114>
- Internet of Things project failure and success 2017: facts and data [WWW Document], 2017. . -SCOOP. URL <https://www.i-scoop.eu/internet-of-things-guide/internet-things-project-failure-success/> (accessed 10.29.17).
- IoT Development Challenges: Why Do IoT Projects Fail?| Upwork [WWW Document], 2017. . Hiring Upwork. URL <https://www.upwork.com/hiring/for-clients/iot-development-challenges-iot-projects-fail/> (accessed 10.17.17).
- IoT Interoperability: Facing The Challenge | Altimeter, 2014. . Prophet Think.
- Jacobson, I., Spence, I., Wei- Ng, P., 2017. Is There a Single Method for the Internet of Things? - ACM Queue [WWW Document]. URL <http://queue.acm.org/detail.cfm?id=3123501> (accessed 10.30.17).
- Javed, B., Iqbal, M.W., Abbas, H., 2017. Internet of things (IoT) design considerations for developers and manufacturers, in: 2017 IEEE International Conference on Communications Workshops (ICC Workshops). Presented at the 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 834–839. <https://doi.org/10.1109/ICCW.2017.7962762>
- Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2009. Systematic literature reviews in software engineering – A systematic literature review. *Inf. Softw. Technol., Special Section - Most Cited Articles in 2002 and Regular Research Papers* 51, 7–15. <https://doi.org/10.1016/j.infsof.2008.09.009>
- Kocher, C., Heron, G., 2014. The Internet of Things: Challenges and Opportunities | Sandhill [WWW Document]. URL <http://sandhill.com/article/the-internet-of-things-challenges-and-opportunities/> (accessed 10.30.17).
- Labs, F., 2015. How IoT is Changing Development, Testing & Software Release Processes [WWW Document]. URL <http://blog.flux7.com/how-iot-is-changing-development-testing-and-software-release-processes> (accessed 10.9.17).

- Liu, T., Lu, D., 2012. The application and development of IOT - IEEE Conference Publication [WWW Document]. URL <https://ieeexplore.ieee.org/document/6291468/> (accessed 10.4.17).
- Liu, Y., Zhou, G., 2012. Key Technologies and Applications of Internet of Things, in: 2012 Fifth International Conference on Intelligent Computation Technology and Automation. Presented at the 2012 Fifth International Conference on Intelligent Computation Technology and Automation, pp. 197–200. <https://doi.org/10.1109/ICICTA.2012.56>
- Lopez, K., 2017. The Role of QA in the DevOps World [WWW Document]. DevOps.com. URL <https://devops.com/role-qa-devops-world/> (accessed 2.7.18).
- Maia, P., Cavalcante, E., Gomes, P., Batista, T., Delicato, F.C., Pires, P.F., 2014. On the Development of Systems-of-Systems Based on the Internet of Things: A Systematic Mapping, in: Proceedings of the 2014 European Conference on Software Architecture Workshops, ECSAW '14. ACM, New York, NY, USA, p. 23:1–23:8. <https://doi.org/10.1145/2642803.2642828>
- Marwah, Mateen, Q., Sirshar, M., 2015. Software Quality Assurance in Internet of Things. *Int. J. Comput. Appl.* 109, 16–24.
- Problems with Traditional Methods of Software Development, 2013. . Eternal Sunshine Mind.
- Sand, B., 2015. IoT Testing - The Big Challenge Why, What and How, in: Internet of Things. IoT Infrastructures, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Presented at the International Internet of Things Summit, Springer, Cham, pp. 70–76. https://doi.org/10.1007/978-3-319-47075-7_9
- Slama Dirk, Puhlmann, F., Morrish, J., M Bhatnagar, R., 2015. Enterprise IoT by. Ignite. Published by O'Reilly Media, Inc..
- Barbara Kitchenham, und Stuart Charters. EBSE 2007-001.Guidelines for performing Systematic Literature Reviews_in Software Engineering.Keele University and Durham University Joint Report, (2007). EBSE Technical Report EBSE-2007-01.
- Software, I., 2016. The Challenges of Quality Assurance and Testing Teams in Software Development [WWW Document]. URL <http://content.intland.com/quality-assurance/the-challenges-of-quality-assurance-and-testing-teams-in-software-development> (accessed 12.6.17).
- Subbarayalu, L., 2017a. Building a Services and IoT Platform with DevOps, Part 1 [WWW Document]. DevOps.com. URL <https://devops.com/building-services-iot-platform-devops-part-1/> (accessed 2.6.18).
- Subbarayalu, L., 2017b. Building a Services and IoT Platform with DevOps, Part 3 [WWW Document]. DevOps.com. URL <https://devops.com/building-services-iot-platform-devops-part-3/> (accessed 2.6.18).
- Subbarayalu, (Lavanya, 2017. Building a Services and IoT Platform with DevOps, Part 4 [WWW Document]. DevOps.com. URL <https://devops.com/building-services-iot-platform-devops-part-4/> (accessed 2.7.18).
- Sulayman, M., Mendes, E., 2009. A Systematic Literature Review of Software Process Improvement in Small and Medium Web Companies, in: Advances in Software Engineering, Communications in Computer and Information Science. Presented at the International Conference on Advanced Software Engineering and Its Applications, Springer, Berlin, Heidelberg, pp. 1–8. https://doi.org/10.1007/978-3-642-10619-4_1

- Suriya, S.M., Manikandan j, Vijay, A., 2017. Interoperability Challenges in IoT. Aricent Blog.
- Kitchenham, 2004. Procedures for Performing Systematic Reviews. Keele, UK, Keele Univ.. 33.
- Taivalsaari, A., Mikkonen, T., 2017. A Roadmap to the Programmable World: Software Challenges in the IoT Era. *IEEE Softw.* 34, 72–80.
<https://doi.org/10.1109/MS.2017.26>
- Tom, C., 2014. The Internet of Things Methodology - Iterating the Internet of Things one Thing at a time. - The IoT Methodology #iotmethodology [WWW Document]. URL <http://www.iotmethodology.com/> (accessed 5.24.18).
- Townsend, J., 2016. The Function of Quality Assurance (QA) with the Internet of Things - CenturyLink Cloud [WWW Document]. URL <https://wwwctl.io/blog/post/qa-with-the-iot/> (accessed 12.6.17).
- Van den Abeele, F., Hoebeke, J., Moerman, I., Demeester, P., 2015. Integration of Heterogeneous Devices and Communication Models via the Cloud in the Constrained Internet of Things. *Int. J. Distrib. Sens. Netw.* 11, 683425.
<https://doi.org/10.1155/2015/683425>
- Virmani, M., 2015. Understanding DevOps bridging the gap from continuous integration to continuous delivery, in: Fifth International Conference on the Innovative Computing Technology (INTECH 2015). Presented at the Fifth International Conference on the Innovative Computing Technology (INTECH 2015), pp. 78–82.
<https://doi.org/10.1109/INTECH.2015.7173368>
- Wallgren, A., 2017. As the IoT expands, agile is the way forward [WWW Document]. TechBeacon. URL <https://techbeacon.com/iot-expands-agile-way-forward> (accessed 1.4.18).
- What is DevOps - Explained [WWW Document], n.d. . New Relic. URL <https://newrelic.com/devops/what-is-devops> (accessed 2.6.18).
- White, G., Nallur, V., Clarke, S., 2017. Quality of service approaches in IoT: A systematic mapping. *J. Syst. Softw.* 132, 186–203. <https://doi.org/10.1016/j.jss.2017.05.125>
- Yeo, K.S., Chian, M.C., Ng, T.C.W., Tuan, D.A., 2014. Internet of Things: Trends, challenges and applications, in: 2014 International Symposium on Integrated Circuits (ISIC). Presented at the 2014 International Symposium on Integrated Circuits (ISIC), pp. 568–571. <https://doi.org/10.1109/ISICIR.2014.7029523>

APPENDIX 1. IoT challenges with links



APPENDIX 2. IoT Implications with links

