Available online
@ https://jjem.jnnce.ac.in
https: www.doi.org/10.37314/JJEM.SP0263
Indexed in International Scientific Indiexing (ISI)
Impact factor: 1.395 for 2021-22
Published on: 08 December 2024

# Review on Impact of IOT on Software Engineering

**Raziya Sultana[1], Suhana Khan [2], Zakiya Farheen[3], Sampath Kumar [4]**

[123] student, [4] Assistant professor, Department of Master of Computer Applications
JNN College of Engineering, Shivamogga

raziyasultana5000@gmail.com, suhanaakhan2002@gmail.com, zakiyafarheen2003@gmail.com

***Abstract***

*The Internet of Things (IoT) has emerged as a transformative technology paradigm, revolutionizing various domains including healthcare, manufacturing, transportation, and smart cities. This paper investigates the profound impact of IoT on software engineering practices and methodologies. We analyze the unique challenges posed by IoT systems, including scalability, heterogeneity, security, and real-time processing requirements. Moreover, we examine how IoT necessitates the evolution of software development processes, architectures, and tools to effectively manage the complexities inherent in IoT deployments. Through a comprehensive review of existing literature and case studies, we identify key research directions and opportunities for advancing the state-of-the-art in IoT-enabled software engineering. Our findings highlight the critical need for innovative approaches to address the intricacies of developing and maintaining robust, secure, and interoperable software systems in the IoT era.*

***Keywords:***

*Authentication, Internet of things, Requirement engineering, Software Engineering, industries, companies.*

## 1 Introduction

The integration of everyday physical objects with the internet enables data gathering and transmission and execution of operations transformative phenomenon that is revolutionizing spheres and people's lives. Owing to connectivity, there are increased devices and data and therefore, software engineers have to design a system that is capable of catering for this ever-increasing number. Middleware and interoperability solutions are needed because of the heterogeneity of IoT, devices, operating systems, and protocols. Engineers, therefore, require new Model driven engineering, code generation, agile DevOps, and other solutions and methodologies to make the Internet of Things a reality. Software developers require adopting these complications to build future dominant strategies of the linked device technology and fully adopting the concept of the Internet of Things.

The paper is organized in the following sections: Section 2 contains the survey of the literature. Section 3 gives brief idea of technique. Section 4 contains Challenges and section 5 explains the Conclusions of the Research work.

## 2 Literature survey

In this review paper [1], the author speaks about the effects IoT has on software business models as well as the importance of the fast shift toward new opportunities and leaving the licenses-or- per-seat business model behind. Zoe Katti et.al.,[2] The presented report aims at exploring how academic works add value to society by conducting an empirical investigation of the authors and a patent search to evaluate the literature on the role of software engineering advancements in practice. [3] The subject of this article is the potential and challenges of IoT application in smart cities with regard to the architectural concept, security, expandability, and information processing. It also emphasizes the importance of the standards, compliance, and several upcoming technologies including the edge analytics and fog computing. [4] The paper reviews up-to-date IoT access control and authentication approaches and proposes a viable solution and conducts a security

investigation on that method against different attacks type including but not limited to replay, eavesdropping and man-in the-middle attacks. [5] With a view to helping elicit future research effort and potentials for practice in IoT software engineering, this paper presents an analysis of the existing approaches and issues. [6] Besides, reflecting upon the IoT software development models and assessing their relevance, advantages, and drawbacks, the article provides knowledge about the strategies and issues, the stages in the lifecycle, the architecture, and testing methodology; such information may be useful to the academic and a practitioner in finding the solution. [7] This article seeks to do a critical analysis of how AI and IoT are being applied in healthcare, and how they present hope for changing the patient care, based on current research, trends, and challenges. [8] Considering resource limitations, decided compatibility, security, grow, architecture layout, interaction conventions, and data management strategies, the article discusses the challenges on accomplishing Internet of Things application systems. [9] This work aims at analyzing the developments of this world technological marvel from the early 1960s up to the present, pointing out the key milestones, technological achievements, and the roles of individuals and organizations in the expansion and establishment of the Internet's norms. [10] This paper stresses on the need for optimality in systems through the illustration of pros, cons, availability, scale, security and ethical aspect of edge computing. [11] This paper precipitates on Internet of Things applications and in sequences of smart cities, agriculture and healthcare looking for service quality improvement, moving on layered architecture and software engineering discipline. [12] This paper focuses on the issues related to the Internet of Things applications and research areas including data management, communication system, security and integration of Blockchain.

## 3 Methodology

The research methodology section outlines the systematic approach undertaken to investigate the impact of the Internet of Things (IoT) on software engineering. Given the transformative potential of IoT, understanding its effects on software engineering practices, tools, and methodologies is critical. This study employs a qualitative research design, primarily focusing on a comprehensive literature review to synthesize existing knowledge and identify key trends and challenges in the field. The following sections detail the research design, data collection methods, data analysis techniques, and the ethical considerations that guided this study.

### 3.1 Before IOT in software engineering

Before the impact of IoT (Internet of Things) on software engineering, the focus was primarily on developing software for traditional computing devices like computers and smartphones. The emergence of IoT introduced a paradigm shift, requiring software engineers to adapt to the challenges of developing and maintaining software for interconnected devices with diverse capabilities and constraints. This includes considerations such as security, scalability, real-time processing, and interoperability across a wide range of devices and platforms.

The impact of IoT on software engineering is significant and far-reaching. Here are a few key points to consider:

- Distributed Systems: IoT introduced the need for software engineers to design and manage distributed systems capable of handling massive amounts of data generated by interconnected devices.
- Data Management: With IoT, software engineering has increasingly focused on developing robust systems for collecting, processing, analyzing, and managing large volumes of data generated by IoT devices.
- Security Challenges: IoT devices often have limited computing resources, making them susceptible to security vulnerabilities. Software engineers now must prioritize security measures such as encryption, authentication, and access control to protect IoT systems from cyber threats.
- Interoperability: Ensuring seamless communication and interoperability between various IoT devices and platforms has become

a key challenge for software engineers. Standards and protocols such as MQTT, CoAP, and Zigbee are essential for achieving interoperability.

- Edge Computing: IoT has led to the rise of edge computing, where data processing and analysis occur closer to the data source, reducing latency and bandwidth usage. Software engineers need to develop applications and algorithms optimized for edge computing environments.
- Scalability and Reliability: IoT systems must be designed to scale seamlessly as the number of connected devices grows. Software engineers need to develop scalable architectures and implement redundancy measures to ensure high availability and reliability.
- User Experience: IoT applications often involve a combination of physical and digital interactions. Software engineers must focus on designing intuitive user interfaces and seamless user experiences that integrate seamlessly with the physical world.
- Lifecycle Management: IoT devices have a longer lifecycle compared to traditional computing devices. Software engineers need to implement strategies for remote device management, firmware updates, and maintenance to ensure the longevity and security of IoT systems.

### 3.2 After the impact of IOT on software engineering

After the impact of IoT on software engineering, several significant changes have occurred:

1. Proliferation of Connected Devices: The number of connected devices has exploded, leading to an increased demand for software engineers skilled in developing applications for IoT ecosystems.
2. Specialized Skillsets: Software engineers now require specialized skills in areas such as embedded systems, wireless communication protocols, data analytics, and edge computing to effectively develop and manage IoT solutions.
3. Shift towards Edge Computing: Edge computing has gained prominence as a critical architecture for IoT systems, leading to a shift in software engineering practices towards developing applications that leverage edge computing capabilities for real-time data processing and analysis.
4. Focus on Security and Privacy: With the growing threat landscape, there's a heightened emphasis on building secure and privacy-preserving IoT systems. Software engineers are now incorporating security measures such as secure boot, data encryption, and device authentication into their designs from the outset.
5. Adoption of New Development Frameworks and Tools: To address the unique challenges of IoT development, new development frameworks, platforms, and tools have emerged. Software engineers are leveraging these resources to streamline the development process and improve the efficiency of IoT projects.
6. Integration with AI and Machine Learning: IoT systems increasingly incorporate AI and machine learning algorithms to derive actionable insights from the massive amounts of data generated by connected devices. Software engineers are integrating these technologies into IoT solutions to enable predictive maintenance, anomaly detection, and other advanced capabilities.
7. Standardization Efforts: Standardization bodies and industry consortia are working to establish common standards and protocols for interoperability, security, and data exchange in the IoT ecosystem. Software engineers play a crucial role in implementing and adhering to these standards to ensure compatibility and seamless integration between different IoT devices and platforms.
8. Emphasis on Sustainability: There's a growing awareness of the environmental impact of IoT devices and systems. Software engineers are incorporating sustainability considerations into their designs by optimizing energy consumption, minimizing resource usage, and promoting the reuse and recycling of hardware components.

### 3.3 Key points

1. Connectivity and Integration: With IoT, software engineers need to develop systems that can seamlessly connect and integrate various devices, sensors, and platforms. This requires expertise in communication protocols, data management, and security.
2. Data Management: IoT generates massive amounts of data that software engineers must handle efficiently. They need to design software systems that can collect, process, and analyze this data in real-time, while ensuring its integrity and security.
3. Security Challenges: IoT introduces new security challenges due to the interconnected nature of devices. Software engineers must implement robust security measures to protect against data breaches, unauthorized access, and potential cyber-attacks.
4. Scalability and Performance: IoT systems often involve a large number of devices and generate high volumes of data. Software engineers must design scalable and high-performance software architectures that can handle the increasing demands of IoT applications.
5. User Experience: IoT devices and applications aim to provide seamless and intuitive user experiences. Software engineers play a crucial role in designing user-friendly interfaces and ensuring smooth interactions between users and IoT systems.
6. Evolving Technologies: IoT is a rapidly evolving field, with new technologies and standards emerging constantly. Software engineers need to stay updated with the latest advancements and adapt their skills to incorporate new tools and frameworks.

### 3.4 Scalability of IOT on software engineering

In the context of IoT and software engineering, **Scalability** refers to an IoT system's ability to manage increasing workloads or its capacity expanded to accommodate growth. This involves not just adding more devices but ensuring that the software infrastructure can efficiently manage increased data volumes, network traffic, and processing demands. Here's an in-depth look at scalability in IoT within software engineering, broken down into key parts:

**Data volume management**

Explanation: IoT devices produce vast amounts of data. Scalability involves ensuring that databases and storage systems can handle
Such Implement hierarchical device management, use robust authentication mechanisms, and ensure efficient user access control systems. Implement performance monitoring tools, optimize code for performance, and utilize caching mechanisms.

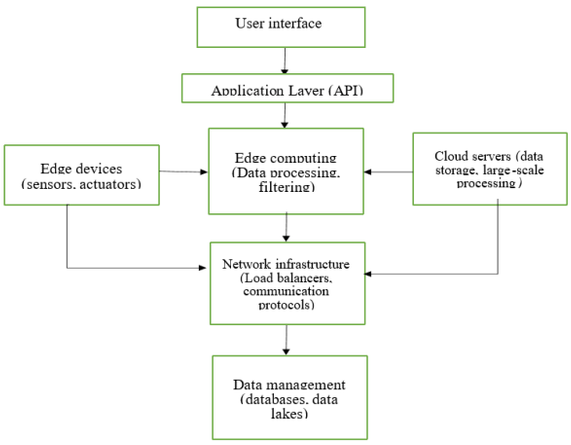


Figure 3.5: Diagrammatic representation of scalability

this influx without performance degradation. Approaches: Use distributed databases like Apache Cassandra or data lakes, and implement data partitioning and sharding strategies.

**Network Traffic Handling**

Explanation: As the number of IoT devices increases, the network must support higher traffic volumes and maintain low latency.
Approaches: Utilize edge computing to process data closer to the source, implement efficient communication protocols like MQTT, and deploy load balancers.

**Processing Power and Computation**

Explanation: Increased data requires more computational power for processing and analysis. Approaches: Leverage cloud computing resources, utilize serverless architectures, and apply parallel processing techniques.

**Device and User Management**

Explanation: Managing a large number of devices and users can be complex. Scalability ensures smooth operation and maintenance.

Approaches: Implement hierarchical device management, use robust authentication mechanisms, and ensure efficient user access control systems.

**Security and Privacy**

Explanation: Scaling IoT systems introduces new security challenges, making it essential to ensure that security measures scale alongside the system. Approaches: Use scalable encryption methods, deploy automated security monitoring tools, and ensure regular scorodites.

**Application Performance**

Explanation: The performance of applications interacting with IoT devices must remain optimal even as the system scales.

Approaches: Implement performance monitoring tools, optimize code for performance, and utilize caching mechanisms.

**3.5 Diagrammatic representation**

Below is a conceptual diagram illustrating the key components involved in scaling an IoT system within software engineering:

**Explanation of Diagram Components**

1. Edge Devices: These are IoT devices like sensors and actuators that generate data and perform actions based on received commands.
2. Edge Computing: Processes data locally or near the data source to reduce latency and network load.
3. Network Infrastructure: Ensures efficient data transmission between edge devices, edge computing nodes, and cloud servers. Load balancers distribute traffic evenly.
4. Cloud Servers: Handle large-scale data storage and processing. Cloud services provide scalable resources to meet increased demands.
5. Data Management: Involves storing and managing the large volumes of data generated by IoT devices. This includes databases and data lakes designed for scalability.
6. Application Layer (API): Interfaces between users and the backend IoT infrastructure, providing access to IoT data and functionalities.
7. User Interfaces: Frontend applications and dashboards that users interact with to control IoT devices and monitor data. By addressing these components and ensuring each layer can scale, IoT systems can efficiently manage increased loads and maintain performance, reliability, and security.

**3.6 IOT system architecture or IOT system framework**

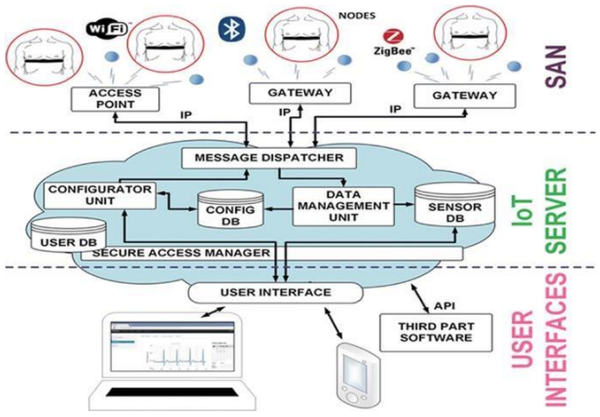


Figure 3.6: IOT system architecture or IOT system framework

The diagram illustrates the working of an internet of things (IOT) system as follows

1. Nodes: Sensors and devices, equipped with communication capabilities (WiFi, Bluetooth, ZigBee), collect data from their environment.
2. Access Points and Gateways: Nodes send the collected data to access points or gateways. These act as intermediaries that manage and direct data flow from the nodes to the central system.
3. Message Dispatcher: The data from access points or gateways is received by the message dispatcher, which is responsible for routing the data to appropriate internal components.
4. Data Management Unit: This unit processes the incoming data for storage, analysis, and

further usage. It ensures the data is properly handled and organized.

5. Sensor Database (Sensor DB): The processed data is stored in the sensor database, where it can be accessed for monitoring and analysis.
6. Configuration Database (Config DB): This database stores configuration settings and parameters for the system and devices, ensuring they operate correctly and can be updated as needed.
7. Configurator Unit: It manages and updates device configurations, ensuring that all devices are operating under the correct settings and protocols.
8. Secure Access Manager: This component manages authentication and authorization, ensuring that only approved users and devices can access the system and its data, thereby maintaining security.
9. User Interface: Users interact with the system through user interfaces, such as web portals or mobile apps, allowing them to monitor data, configure settings, and manage devices.
10. API and Third-Party Software: The system provides APIs for integration with third-party software, enabling external applications to access and utilize the IoT system's data and functionalities for extended capabilities and services.

**Challenges**

1. Security and Privacy: IoT devices are prone to cyber-attacks and ensuring data privacy is complex.
2. Interoperability: Diverse communication protocols and lack of universal standards complicate device integration.
3. Scalability: Managing vast amounts of data and increased network traffic requires scalable solutions.
4. Resource Constraints: Limited power and computational capabilities of IoT devices pose challenges.
5. Reliability: Ensuring consistent performance and handling device failures are critical.
6. Data Quality: Integrating and standardizing heterogeneous data from various devices is difficult.
7. Real-Time Processing: Processing data in real-time for applications like autonomous vehicles is challenging.
8. Development Complexity: Rapid technological changes require continuous learning and adaptation.
9. User Experience: Designing intuitive interfaces and ensuring user trust are essential.
10. Regulatory Compliance: Adhering to different regional regulations and standards is necessary but complex.

## 5.conclusion

The integration of IOT in software engineering is transforming the industry, offering enhanced automation and real-time data capabilities. However, this transformation comes with significant challenges such as security concerns, interoperability issues, and scalability demands. Addressing these requires scalable, secure, and user-friendly solutions. By tackling these challenges, the software engineering field can fully leverage the potential of IoT, driving innovation and improving efficiencies across various domains. Collaboration and ongoing research are key to overcoming these hurdles and realizing the full benefits of IoT.

## Reference

1. Krzysztof Wnuk and Bhanu Teja Murari, "The impact of Internet of Things on Software Business Models", 7th international conference, ICSOB Ljubljana, Slovenia, vol.7, No.94-108, 07 June 2016.

2. Zoe Kotti ET.AL., "The impact of Software engineering research in practice", IEEE Transactions on Software Engineering , Vol.49, No.04, 01 April 2023.

3.Jun Zhou ET.AL., "Security and privacy for Cloud-Based IOT: Challenges, Countermeasures and Future Directions", IEEE

Communications Magazine, vol.55, issue.1, No.26-33, 19 January 2017.

4.Jing Liu ET.AL., "Authentication and Access Control in the Internet of Things", 32nd International Conference in Distributed Computing Systems Workshops", vol.7(4), No.588-592, June 2012.

5.Mahdi Fahmideh ET.AL., "Software Engineering for Internet of Things: The practitioners' perspective ", IEEE Computer Society, 28 April 2021.

6.Shereen Ismail and Diana W. Dawoud, "Software Development model for IOT," IEEE 12th annual computing and communicating workshop and conference (CCWC), January ,2021.