International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Key Metrics Identification of Distinct Process Models to Develop an Iot Based Systems

Satyanandaram N, Smitha Chowdary CH

Abstract*: The globe is encompassing gravitate automation. Many organizations turning the theoretical aspects into practical implementation by converging technologies. Embedded systems considered the core of sensor-based technology. Bundle of built-in configuration required to interact devices. Productivity software confederate to communicate with the integrated systems over the web. Process models furnished strategies to implement systems. Process models like a classical waterfall, Spiral, XP, Agile, RUP, Prototype, and Scrum which provides the blueprint for developers. The paper investigating the variations among process models and integrated system development. Discrete factors have to be considered an excerpt of the process model to an integrated system. Process models can be suitable to implement small scale and medium scale systems but implementing sophisticated systems deals with various critical factors. The non-functional attributes of development tools and deployment environments are the key components in system development. Since the cloud is the component of IoT and cloud providing solutions as per pay and use policy. Identifying process model for IoT systems depends on application nature, development and deployment environments. IoT systems require real-time operating systems or devices to interpret code in it. The probable scrutiny is given information about the agile methodology. The research involving with identification of key metrics of various process models to develop IoT based systems. The paper presuming the external factors not involving with the process mode. Identified key factors from agile model to develop medium level integrated systems.

***Keywords*: Agile, Embedded Systems, IoT, Micro controllers, Process models, Systems engineering.**

I. INTRODUCTION

In this technological era, everyone wants to get easier and convenient services with the help of technology. The continuous efforts of the engineers and stakeholders are discovering directions in IoT. Convergence of technologies providing digitalized and automated services with devices. But the implementations are very complicated and highly expensive [1].

Manuscript published on November 30, 2019.
* Correspondence Author
Mr. Satyanandaram nandigam*, Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Green Fields, Guntur District, Vaddeswaram, Andhra Pradesh, India
Dr. Chaparala Smitha Chowdary, Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Green Fields, Guntur District, Vaddeswaram, Andhra Pradesh,

IoT hosts devices with integrated systems, device software, network connectivity, and sensors. An integrated system is a tool incorporating executable code with hardware [2]. IoT consists of hardware components like microcontrollers, wireless, Bluetooth, and GSM modules. IoT deals with device broadcasting over data networks. IoT consists of embedded systems, application interface, cloud, database, and communication channel [3].

- Embedded systems: The programmable device implemented with an assembler interpreted code or compiler executed code in an electronic machine. Hardware components are microcontrollers, Processor, power supply, memory, timers, and counters[4]
- Applications: over the network, integrated systems operate with an application. Data interchange with integrated systems which are under IoT requires a user interface in terms of mobile/web-based application.
- Cloud interface: IoT systems bring about large data and cloud come up with warehouse destination. Cloud approaches are available for various expansion and distribution domains. The convenient expansion services and servers obtainable under the cloud domains. Under the pay to use scheme cloud furnishing the assistance.
- Database: Every IoT device requires a database to maintain data being generated. Data clusters being maintained by IoT servers.
- Communication methods: To link up the integrated systems over the internet, it requires communication devices like Bluetooth, Wi-Fi, and GSM modules. By using one of the modules, an HTTP server has to initialize for interacting with the application over the network.

The favorable results of the advancement of a unified system build upon its software and hardware incorporation. Developers implements quality applications by using process models. Electrical engineers design for both digital and analog devices. Manufacturers assemble physical devices according to application notes given by the developers. Here the process involving with integrating effective design processes together with practical software techniques in order to implement quality integrated systems. During the implementation of smaller integrated systems, no design model principles have to follow, simply coding can be sufficient description.Software engineering derives efficient access to develop applications. It is providing an organized set of past experiences. These are arranged as guidelines and methodologies [5]. Generally, to implement a smaller system requires programming.

To implement a bigger system should follow the software engineering technics. By incorporating the technics one can develop bigger software products and achieve quality systems. Product customization can be controlled by size and complexity. The need of the software engineering paradigm depends on the size and complexity of the system [6].The software engineering process includes feasibility study, requirements gathering and analysis, design, coding, testing, and maintenance. Systems engineering path starts with contemplation and decomposing. The path provides simpler ways to structure and reduces complexities by using abstraction and decomposition. Abstraction can be simplified by not considering the irrelevant details. Decomposition objective is subdividing difficulty [7]. The subdivided pieces develop accordingly.

II. LITERATURE REVIEW

A. Embedded System

An Embedded system [8] accomplishes a particular job after merging the instructions with the device. Embedded or Integrated systems are the consolidation of personalized devices and customized software which executes pre-defined tasks. Embedded systems having a wide range of applications like navigation tools, video cameras, cell phones, cars, electronic devices [9], etc… Embedded systems are having small software programs implemented as functions which never able to modify if the product was being deployed. Microcontrollers configured with the system to do a distinct activity. Generally, sophisticated systems implementation involving with intricacy. Systems evolution involving with difficulties and crave the need for hardware and software co-design, integration and testing the final system [10]. Embedded system merging the hardware with an operating system. system behavior has to be designed. The process models can be applied to implement unified systems.

B. Different types of Process Models

Process models are a purely illuminating portrayal of the evolutionary activities carried out during the software life cycle. the process models breach the system into collective modules and defined the procedure to implement a specification and get the solution. The association of specifications will setup the results. Process model defines the order of actions to be carried out. It maps the steps accomplished on a system from its inception to exit [11]. Lack of process models, the product would not be in an organized and disciplined mode. Process models have to introduce during system evolution. Those are given below description

- Classical waterfall model
- Iterative waterfall model
- Incremental model
- Spiral model
- Prototype model
- Agile model
- V Shaped Model

Classical waterfall model: The model considered a theoretical way of developing software. It considered the origin of all other process models. it provides the successive and cumulative development of the system [12]. The model depicted the error-free system will implement by developers. The model divides the system evolved into the following phases.

- Feasibility study
- Requirements analysis and specification
- Design
- Coding and unit testing
- Integration and system testing
- Maintenance

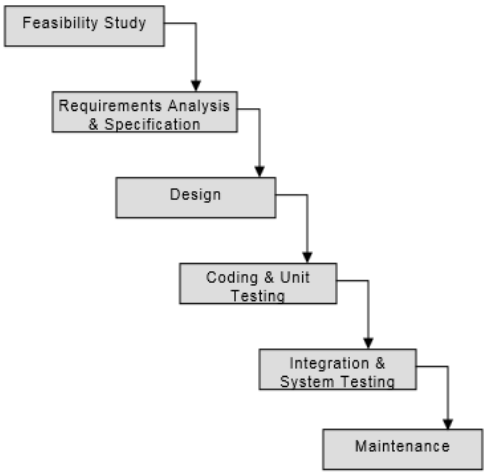


Fig. 1.Classical waterfall model.

Iterative waterfall Model: The model is an advancement of the above model. The next level will start after getting the feedback from the customer. In the process of getting a final version, the smaller versions have been implemented and integrated with each iteration [13].

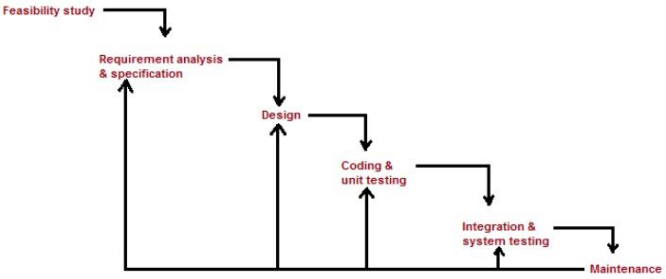


Fig. 2.Iterative waterfall model

Incremental model: The model depicts the process that requirements are split into multiple objectives. The system will be implemented as a series of variants, hence the current variant will upgrade with the previous variant specifications [14].

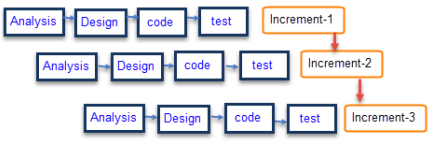


Fig. 3. Incremental Model

Spiral Model: The model depicted as a curlicue. It includes the linear, repetitious essence of the mock-up. The model accepts the client assessment and repetition and risk analysis continue throughout product development.

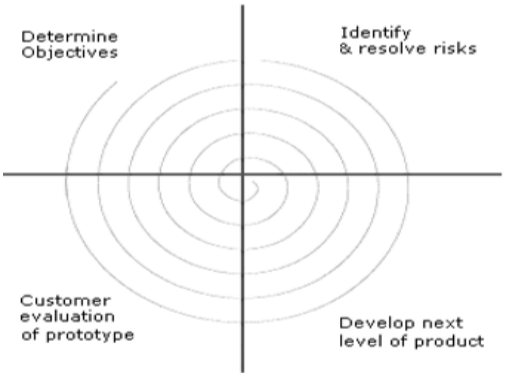


Fig. 4. Spiral Model

Prototype model: The model handover the trinket system in each iteration. If customers are non-technical and usually unclear about the requirements this model navigating them. Missing functionalities and errors will discover easily [15].

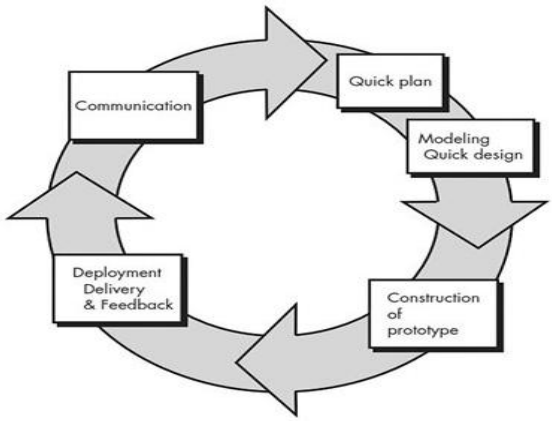
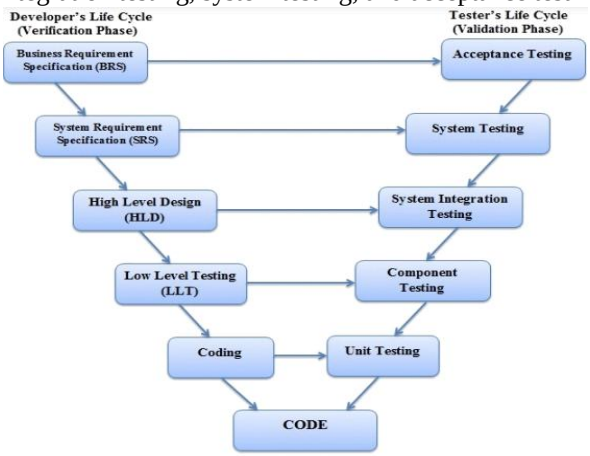


Fig. 5. Prototype model

V Shaped Model: The model depicts the subsequent manner of procedure execution. Each phase associating with testing. The main aim of this model is the verification and validation of requirements. Under the verification, the developer will be associated with requirements analysis, system design, architectural design, module design, and coding phase. Under the validation, the tester will be associated with Unit testing, integration testing, system testing, and acceptance testing



.

Fig. 6. V-Shaped model

C. Different types of Embedded Systems

Developing an integrated system is a key phase in IoT. Many industries developing the embedded systems. A programmed or non-programmed electronic device which uses to operate and integrated to perform a single task or multiple tasks is called an embedded system. IoT applications can be used to monitoring and controlling safety and mission-critical systems [24]. Based on the performance of the microcontroller embedded systems has divided into three categories [16].

- Small scale Embedded system
- Medium scale Embedded system
- Sophisticated Embedded system

Small Scale Embedded System: These systems can operate with a small battery and uses an 8-bit or 16-bit microprocessor. To develop these systems will require writing code in assembly language or embedded C programming. The compiler translates the embedded C code into HEX code. The assembler converts the instructions into machine code. The hardware device will take the hex instruction and write it into the ROM of the processor [17].

Medium Scale Embedded Systems: These systems can operate with 16-bit or 32-bit microprocessors. To develop these systems requires the collaboration of hardware devices with software programs. During the development of these systems, it requires programming tools like compilers, assemblers, debuggers, Visual C++, Source code engineering tools, simulator, and IDEs. Developers have to provide an application programming interface (APIs) to interact with these devices on Real-time operating systems (RTOS) for controlling the microcontrollers [18].

Sophisticated Embedded systems: The integrated system which executes extensive work on multiple 32-bit or 64-bit microcontrollers are known as sophisticated embedded systems. Implementation of these systems involving with more complexities since it requires complicated and costly designed tools. These systems require its real-time operating system (RTOS) on critical time-bound applications to perform the large computations [19]. These systems are used to develop devices like smartphones, multi-media systems, etc…

D. Process Models for Embedded Systems

To implement integrated systems, pragmatic design processes and rational techniques have to include. Based on the size and intricacy, software development changes from simple coding to software engineering [20]. The team has to follow the given phases during implementation.

- Diagnose the system: find out the process and diagnose the instructions and results from the process. Investigator collects the requirements and should resolve inconsistencies, anomalies, and incompleteness by discussing with the end-user or client.
- Requirements gathering: Investigators will gather all information regarding the system to be implemented from the users by conducting discussions and interviews. Based on the requirements the problem statements will be generated. After gathering the requirements, developers will find out the technical feasibility,functional requirements and non-functional requirements.

- System specification: A precise way of explanation will be developed in terms of an algorithm. The developer has to prepare all preconditions, post conditions, success scenarios, and alternate approach.
- Design Phase: System designers have to prepare the theoretical model. The practical division of the structure will be accomplished. Each function considered a data alteration block. A data flow diagram shows the stratified pictorial model of the data process between different tasks that the system executes. DFDs represent the data alteration among the blocks.
- Implementation: Based on the decomposition notion the scheme exertion will be done. Simulators will be introduced to check the correctness of the system. The subcomponents will be exertion and executed on the simulator and this allows the rapid debugging and success of the product development. To get the machine code on the target system, source code has to convert by cross compilers and cross assemblers.
- Testing: Testing: system performance has to be evaluated. During the evaluation, it requires the debugging and testing the entire system. Memory management, stability, execution speed, and accuracy will be tested during the performance optimization.
- Maintenance: deploying the system and monitoring, rectifying faults, adding a new module, porting the system into a server or operating systems. This phase encourages satisfying the non-functional requirements, adding new features and supports iterations all along the system life cycle.

III. CONTINGENT ANALYSIS OF PROCESS MODEL

To implement any system developers have to select the process model according to the size of the system being developed. They have to scrutinize the following suggestions.

- Risk analysis
- System complexity
- Cost control
- Unknown technologies
- Short term schedules
- Available Technical skill expertise
- User interactions during the development
- Project documentation
- Stakeholders observation
- Project management

After comparing the above key points with all process models they will select the suitable life cycle model for their system.

The comparison metrics as described in Table 1.

Table 1. Key metrics of process model for Integrated Systems development

	Waterfall model	**Incremental model**	**Spiral model**	**Prototype model**	**V-Shaped model**	**Agile model**
Risk Analysis	Poor	Poor	Excellent	Average	Good	Good
Complex System	Good	Good	Excellent	Excellent	Good	Poor
Cost Control	Poor	Poor	Good	Poor	Poor	Excellent
Unknown Technologies	Poor	Good	Excellent	Good	Poor	Poor
Short term schedule	Poor	Excellent	Excellent	Excellent	Excellent	Depends on Project
Technical skills expertise	Required	Required	Not required	Medium	Medium	Required
User Interaction	At the beginning	At the beginning	High	High	High	High
Documentation	Excellent	Excellent	Excellent	Good	Good	Average
Stakeholders observation	Good	Good	Excellent	Excellent	Good	Excellent
Project management	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent

As per the metrics, the waterfall model encourages project management, stakeholder's observation, and documentation. If the requirements are stable, it is suitable for implementing complicated systems. The incremental model focuses on short term schedule, project management, stakeholders' observation, and developer can learn new technologies to develop the system. It is not suitable for small systems and requires good design planning. The spiral model is ideal for all metrics. It is not suitable for small and non-risky systems. No time and cost limitations. Due to the unstable requirements Project management might more complex and project duration might be extended. The prototype model is good for complicated systems, short term schedules, client interactions, higher outputs, and preliminary visibility. If the requirements and technical issues are not clear you can select a prototype model. V-shaped model is easy to understand and it focuses on verification and validation. It is suitable for critical systems and the developer can see the early production. It provides project management and encourages the user interaction. The agile model encourages the user interactions project management, stakeholder's observation. This is the most realistic approach to develop a system and flexible for developers.

IV. AGILE PROCESS MODEL

In traditional system development, communication between clients and customers is very less. Identifying requirements and processes are fixed before analysis and design. Agile software development is "An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative and evolving manner by self-organizing teams within an effective governance framework with "just enough" ceremony that produces high quality solutions in a cost-effective and timely manner which meets the changing needs of its stakeholders"[21]. Each phase follows the continuous testing and evaluation. The process encourages consistent team collaboration. Teams provide a rapid response to a requirement change. Customer involvement is important during system development. Frequent delivery of the working system will be given to the client.

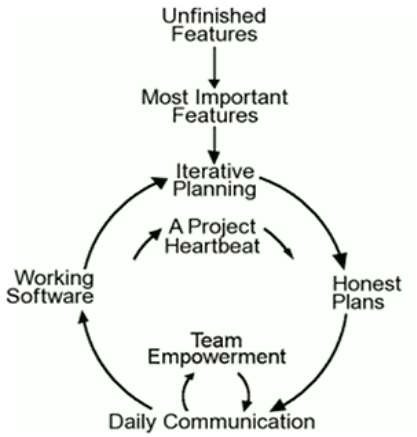


Fig.7. Agile Process model

V. AGILE PROCESS MODEL FOR MEDIUM SIZE APPLICATIONS

Identified key points in Agile process model during the system development. While developing medium size applications, agile process model might be suitable.

Table 2. Key Observations of Agile Process model

Idea	Description
Customer feedback	Regular monitor and feedback by the customer will resolve the intricacy
Incremental Product development	At every iteration, the plan for the next module of the software will be developed.
Motivated people	The group work will be recognized and the group will be given priority. They will be given a good environment, support, and trust
Accept changes	The model encourages the change of requirements in each iteration
Avoid complexity	While developing the system, the process focuses on simplicity. During the interactions with the customer, the complexity will be eliminated from the system
Pre Planning	Embedded system development involves integrating various modules and agile encourages team integrity during the development
Team Integrity	Embedded system development involves integrating various modules and agile encourages team integrity during the development
Serialized processes	Based on the disintegration path the modules will be broken into small increments and follows the iterative approach.
Programming paradigm	any programming language approach can be suitable, but having object-oriented approaches

VI. SOPHISTICATED EMBEDDED SYSTEMS – PROCESS MODEL

- **Key Observations not involving with process models:** To implement the sophisticated systems involves with hardware and software complexities. SES (Sophisticated Embedded systems) built to run on real-time operating system. They have to react on dynamic data. SES uses the embedded hardware and software. The low level software would be operating system and device driver software. The components might be available for free or we need to purchase the licenses [22]. Few process models are unable to use to development of SE systems due to limited availability of tools. Many SE systems have to create their own tools for debugging and testing.

Developing the SE systems involving with convergence of various technologies sometimes may not go according to selected process model. Dynamically the workflow might be change according to availability of development tools.

VII. CONCLUSION

The selection of process models including many factors like application nature, time complexity, cost limitations [23], system quality, and availability of technologies so on. After analysis of process models which have been discussed above, the agile process model is suitable for small and medium-range integrated system expansion. Expansion of sophisticated systems involving with non-functional requirements of hardware, real-time operating system, advanced microcontrollers, debugging and testing tools, integrated tools. Cloud technology providing solutions for development and maintenance. System development tools like IDE's, servers, third party plugins, frameworks and database support will be offered by Cloud vendors. During the system implementation, development tools and hardware components will play crucial role to select process model.

REFERENCES

1. Charara, S. (2018, December 18). IoT challenges: what are they and how are they slowing adoption? Retrieved from https://www.raconteur.net/technology/iot-challenges.
2. Pavan Kumar, T., Hemanth Krishna, R., Sai krishna, M., & Meghana, J. (2018). Smart home system based on IoT. International Journal of Engineering and Technology(UAE), 7(2.8 Special Issue 8), 500-502. Retrieved from www.scopus.com
3. Desai, D. R. (n.d.). Internet of Things. Retrieved from http://drrajivdesaimd.com/2016/07/19/internet-of-things-iot/.
4. Muhammad Ayub Sabir, Muhammad Sheraz Arshad Malik, Fatima Ashraf and Rubab Rasheed. (2018, August). Title. *Embedded IOT System: Software and Security Attacks* [English]. Volume(18), No.8. Available: http://paper.ijcsns.org/07_book/201808/20180810.pdf
5. Pressman, Roger S. Software engineering : a practitioner's approach / Roger S. Pressman. — 7th ed
6. Rohit Sharma et al,Future Challenges and Opportunities of Systems and Software Engineering Processes. (2011). International Journal of computer Science and Information Technologies, Vol. 2 (5), 2176–2179
7. Bell, Doug, 1944- Software engineering for student/Douglas Bell. -- 4th ed. p. cm. Rev. ed. of: Software engineering. 2000. ISBN 0-321-26127-5
8. Harpreet Kaur1 , Jagatjit Singh2. (2013, December). Title. *Review on Embedded System* [English]. Volume 1 (9), Available: https://pdfs.semanticscholar.org/d5b8/5edc3e4573277082bf994d244543a5ac5579.pdf
9. Agarwal, T., & Edgefx Technologies Pvt Ltd. (2014, June 7). Embedded Systems Role in Automobiles with Applications. Retrieved from https://www.edgefx.in/importance-of-embedded-systems-in-automobiles-with-applications/.
10. Micaela Serra,Hardware/Software Codesign. (n.d.). Retrieved from https://webhome.cs.uvic.ca/~mserra/HScodesign.html
11. (PDF) Analysis of various Software Process Models. Available from: https://www.researchgate.net/publication/316510707_Analysis_of_various_Software_Process_Models [accessed Nov 05, 2019].
12. Naga Malleswari, D., Kumar, M. P., sathvika, D., & Kumar, B. A. (2018). A study on SDLC for water fall and agile. International Journal of Engineering and Technology(UAE), 7(2), 10-13. doi:10.14419/ijet.v7i2.32.13516. Available: https://www.sciencepubco.com/index.php/ijet/article/view/13516/5398
13. Kazim Ali. (2017, Jan). *A Study of Software Development Life Cycle Process Models*. International Journal of Advanced Research in Computer Science [English]. Volume 8(1), paging if given. Available:https://www.ijarcs.info/index.php/Ijarcs/article/download/2844/2827
14. Sommerville, Ian Software engineering / Ian Sommerville. — 9th ed
15. Margaret Rouse, *Prototyping Model*, Retrieved November 13, 2019, fromhttps://searchcio.techtarget.com/definition/Prototyping-Model
16. Yuvayana, and Yuvayana Admin. "Classification of Embedded System with Details." Engineers Portal, 24 May 2017, er.yuvayana.org/classification-of-embedded-system-with-details/
17. Maxfizz.com. (2019). Types of Embedded Systems. [online] Available at: https://www.maxfizz.com/types-of-embedded-systems/ [Accessed 13 Nov. 2019].
18. Devine, J., Finney, J., Halleux, P. D., Moskal, M., Ball, T., & Hodges, S. (2019). MakeCode and CODAL: Intuitive and efficient embedded systems programming for education. Journal of Systems Architecture, 98, 468–483. doi: 10.1016/j.sysarc.2019.05.005
19. Embedded Systems: Introduction. (n.d.). Retrieved from https://nptel.ac.in/content/storage2/nptel_data3/html/mhrd/ict/text/108102045/lec1.pdf.
20. Users.ece.utexas.edu. (2019). Chapter 7: Design and Development. [online] Available at: http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C7_DesignDevelopment.htm [Accessed 14 Nov. 2019].
21. Scott W. Ambler. (n.d.). Disciplined Agile Software Development: Definition. Retrieved from http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm
22. Micaela Serra,Hardware/Software Codesign. (n.d.). Retrieved from https://webhome.cs.uvic.ca/~mserra/HScodesign.html
23. Prasada Rao C., Siva Kumar P., Rama Sree S., Devi J. (2018) An Agile Effort Estimation Based on Story Points Using Machine Learning Techniques. In: Bhateja V., Tavares J., Rani B., Prasad V., Raju K. (eds) Proceedings of the Second International Conference on Computational Intelligence and Informatics. Advances in Intelligent Systems and Computing, vol 712. Springer, Singapore
24. Bhanu, Smt. J. Sasi et al. "AN EFFICIENT ARCHITECTURE FOR SEMANTIC EVOLUTION OF EMBEDDED SYSTEM." (2015).

AUTHORS PROFILE

Mr. SATYANANDARAM NANDIGAM, Research Scholar, Department of CSE, Koneru Lakshmaiah Education Foundation, Green Fields, Guntur District, Vaddeswaram, Andhra Pradesh, India , research fields are Software Engineering, Cloud Computing, Internet of Things and Databases

Dr. CHAPARALA SMITHA CHOWDARY, Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Green Fields,GunturDistrict, Vaddeswaram, Andhra Pradesh, India , research fields are Software Engineering, Cloud Computing, Internet of Things and Databases.