



Original Article

Understanding the Use of Agile Development Practices Alongside Quantum Computing in Modern Engineering Projects

Prabu Manoharan¹, Muzaffer Hussain Syed², Uday Kumar Ragireddy³, Prasanth Varma Addepalli⁴, Sridhar Reddy Bandaru⁵, Dhuli Shyam⁶

¹Information Technology, Bourns Inc, HRIS Manager, California, USA.

²Sr Software Developer, Visual Technologies, Plano, TX.

³Sr Technical Program Manager, Vdrive IT Solutions, Inc, Richardson, Texas.

⁴Data Engineer II, Cox Automotive Corp Svcs LLC, Atlanta, Georgia.

⁵Program Management, IT, Microsoft, Senior ACE Engineer, Redmond, WA.

⁶Business Application, IT, Nagase Holdings America Corp, Manager, Application & Software Development NYC, NY.

Abstract - *The understanding of the Agile development practices and quantum computer are becoming more and more a requirement as the engineering projects of the day are growing extremely complicated and dynamic. Agile methods that are familiar to facilitate an iterative cycle, cooperate and adaptability are inherently appropriate to the experimental and unpredictable world of quantum software development. Quantum computing (QC) that is grounded on the principles of superposition and entanglement is the origin of novel computational paradigms that require new programming and continuous fine-tuning. In addition to satisfying this need for brief development cycles and quick feedback loops, other Agile methods like Scrum, Kanban, and Extreme Programming (XP) also demand rigorous planning. On the other hand, quantum software engineering is faced with specific issues such as the unskilled personnel, testing issues, and hardware limitations. In that case, the Agile methodologies are salvaged and assist in the breaking down of complex and quantum tasks into manageable units and result in a continuous improvement. All in all, standardization of Agile practices and quantum computing does not just assist in the innovation, but also in reduction of the risk of development and speeding up the development of the practical quantum solutions to the current engineering projects.*

Keywords - *Quantum Software Engineering, Extreme Programming, Quantum Computing, Software Development, Agile Frameworks, Quantum Programming.*

1. Introduction

Agile software development has completely revamped the software industry and it is currently one of the most powerful theories in the contemporary engineering [1]. It provides organizations with a channel through which they can operate far much faster, flexibly and adaptively in variable environments [2]. Agile approaches, i.e. the ones based on the ideas of the Agile Manifesto, attach emphasis to the iterative development, constant provision of functioning software and communication with the customer to a large degree [3]. These are the underlying aspects of different Agile practices. To illustrate, Scrum minimizes documentation in contrast to the traditional approaches, and it still performs quick iterations; Iconix employs a semi-agile approach and minimizes the number of artifacts, still retaining the significant modeling elements; Extreme Programming (XP) is all about taking care of the quality of the software and being responsive to the ever-changing customer requirements; and lastly, Crystal puts people first and the interaction first as it uses light processes.

Quantum computing (QC) has arisen as a transformative novel way of computing at the same time that Agile approaches have evolved. The quantum computers are able to leverage quantum principles like superposition and entanglement to tackle some complex problems tenfold quicker than the conventional machines [4]. QC is however not a universal accelerator; although it can be used to solve some problems that would be difficult in classical computing to compute in the lifetime of the universe, other problems demonstrate little or no improvement in QC. Regardless, QC has the potential to transform several disciplines, including optimisation, cryptography, materials research, and M.

These areas, when collided, have resulted in the creation of QSE that is defined by a combination of good engineering with the creation, utilisation, and support of quantum software [5]. The main concern in QSE is how to keep classical software engineering methods applied to the quantum world and also develop new methods to meet the quantum-specific constraints [6]. The future quantum applications predominantly rely on hybrid models that integrate classical and quantum elements, and thus, QSE required to assure impeccable interoperability between the two environments. It plays a critical part in the process of designing modular, scalable, and integrable quantum services using the principles of Service-Oriented Computing (SOC) and implementing them utilizing strong Service Engineering techniques.

1.1. Organization of the paper

This paper is organization as follows: Section II highlights Agile development methodologies and important foundations. Section III introduces key principles and technology of quantum computing. Section IV examines the combination of Agile techniques and quantum computing in engineering. Section V includes a literature review on relevant topics, and Section VI closes the study with suggestions for further research.

2. Agile Development Practices

Agile techniques are defined as a set of procedures and methods that adhere to specific ideals and tenets. They both have the characteristic of incremental, iterative development, which produces high-quality products, adapts quickly to changes in requirements, and meets client needs. Agile methodologies emphasize client involvement over contract discussions, working software over long documentation, individuals and interactions above procedures and technology, and mobility over rigidity [7]. Their purpose is to improve business and generate new ideas to suit consumer demands by quickly defining resources when requirements or technology change, as well as responding quickly to market changes or poor customer connections. The most extensively used agile methodologies in the sector include XP, Scrum, the Crystal Family, and DSDM.

2.1. Agile Manifesto and Principles of ASD

The ASD manifesto is described as follows:

While the objects on the right have worth, they appreciate the items on the left more. Instead of analyzing whether the stated concept is good or wrong, which might, in reality, be situation-dependent, this section describes the principles behind ASD (Awasthi, Rai). The 12 concepts expressed in the Agile Manifesto are formally written below:

- The primary objective of agile software is to satisfy clients by consistently and early product delivery.
- At every level of the Agile processes, modifications to the software requirements are always accepted for the customer's competitive advantage.
- Although it usually takes a few weeks to a few months to produce functional software, the shorter time frame is recommended.
- Throughout the project, software developers work every day with clients or businesses.
- Projects are designed around experienced and devoted staff. Since they are given the required environment and support, it is assumed that they finish the task.
- The most effective and efficient way to convey data with and within a development team is seen to be face-to-face contact.
- Software functionality is meant to be the main objective of advancement.
- Agile methodologies are intended to help sustainable development.
- Continuous attention to technological quality and creative design is regarded to increase agility.
- It is claimed that simplicity, the skill of optimizing the amount of effort not done, is vital.

2.2. Key Frameworks of Agile Development

This section describes essential Agile development frameworks, Scrum, Kanban, and XP, which are commonly used to manage and optimize software development processes. Scrum focuses on teamwork and time-boxed sprints and the role definition. Kanban is based on the visualization of the work processes, as well as the restriction of the work-in-progress in order to deliver it just-in-time. XP encourages technical excellence through approaches like pair programming, continuous integration, and test-driven development. When combined, these frameworks offer flexible and effective techniques for contemporary software engineering.

2.2.1. Scrum

Scrum is a framework that enables Agile teams to quickly and clearly develop and maintain composite products. In any event, fully mastering and applying the technique is still challenging. Additionally, Scrum encourages colleagues to apply a diversity of methods and approaches. Decisions that outline the events, elements, and histories as well as the connections and interconnections among them form the foundation of Scrum. Priority is given to the internalization of groups that are not offended by strangers [8]. Scrum-using agile workers also want a team that is both big enough to finish a lot of work in one sprint and sufficiently small to be flexible. For example, it is regarded that the fairest size for an advancement group is five to eleven co-workers, taking into consideration the Product Owner and Scrum Master.

2.2.2. Kanban

In 2003-2004, David J. Anderson introduced the Kanban technique to Agile to help a struggling small Microsoft team. Anderson, however, only presented the Kanban information to the Software Development team in 2010, six years later. In essence, Kanban aims to safeguard the team from the never-ending work that an administration assigns while acknowledging a flexible pace of continual growth. This contrasts with other Agile approaches, where there is essentially no defence against change [9]. The JIT principle, which argues that the team should not develop software features ahead of demand, is the core of

the Kanban technique and is particularly significant for structuring software development. To minimise waste and keep it as near to zero as feasible, the team is not obliged to create or offer superfluous highlights.

2.2.3. Extreme Programming (XP)

XP is a well-known Agile methodology. It has recently been established to be incredibly successful in several organizations of all sizes and in preparations across the globe [10]. Kent Beck, Ward Cunningham, and others developed the minor plan technique known as XP. XP has aroused tremendous intrigue and no little bit of controversy after high profile triumphs. One of the main aspects of the intrigue lies in the fact that XP is a practical approach to change management. The major practices are upfront combination of programming with writing tests, refactoring, and refactoring. The main standards are incremental and iterative improvement, work with the simplest set up, superfluous documentation elimination, and shared code ownership.

3. Understanding Quantum Computing Principles

Quantum computing relies on basic notions of quantum physics that, in particular, give significantly greater computational capacity as compared to traditional computers. Knowledge of these concepts is the key to comprehending the use of quantum technologies in future engineering projects. The basic concepts are the following:

- Superposition: Qubits are capable of existing in a superposition. Unlike qubits, which can have both 0 and 1, classical bits can only have one of two values. This significantly speeds up some types of processing by enabling quantum computers to investigate a large number of options simultaneously.
- Entanglement: Even if two qubits are physically separated, their states become so coupled when they are entangled that it is impossible to explain one qubit's state independently of the other. This characteristic is useful for quantum algorithms and quantum communication.
- Quantum Interference: Quantum algorithms can enhance the likelihood of accurate outcomes and decrease the risk of incorrect ones due to quantum interference. The property is applied in QC to boost algorithm efficiency.
- Quantum Measurement: Upon measurement, a qubit decreases from a superposition to one of the potential outcomes (0 or 1). Measurement contributes a crucial indeterminacy to QC but permits relevant information to be obtained.

3.1. Key Quantum Principles: Superposition and Entanglement

One key component of quantum algorithms has been established to be the capacity of quantum systems to reside in coherent superpositions of eigenstates. Nonetheless, superposition of eigen solutions is not inherently quantum and is a typical feature of many traditional wave equations. It would be really helpful if Jozsa and Ekert debated this topic. Using a multi-qubit system, the authors apply a fixed two-qubit quantum gate U to any pair (i,j) of qubits [11]. Following n steps, a probability distribution $P_n = \{p_n(0), p_n(1)\}$ is used to measure qubit 1 and yield 0 or 1. The outcome can be obtained in time on by applying this procedure to a physical quantum system. They then examine how the same probability distribution, *c a. p, cap P, s u b, n*, may be sampled using traditional methods. They demonstrate that $O(2^n)$ coefficients must be stored in order to describe this process in a classical simulation. According to what they state, "The exponentially increasing amount of data that must be processed at each stage causes a classical simulation to slow down exponentially over time." Therefore, $O(2^n)$ time steps are needed for a classical simulation, whereas $O(n)$ time steps are needed for a quantum simulation.

3.2. Current Quantum Technologies

This section presents an overview of the technologies that underpin QC. Its investigation includes quantum hardware platforms, which constitute the foundation of QC [12]. They scrutinize quantum API interfaces and software development packages, the gateway to the quantum algorithm design, and explore quantum cloud services, which allows the ease of access to quantum resources.

3.2.1. Quantum Software Development Kits

IBM and Microsoft, among others, provide integrated development kits such as the Qiskit and Quantum Development Kit that have tools, libraries and simulators to design, test and optimize quantum algorithms and circuits [13]. They are software development kits (SDKs) which offer quantum algorithms toolkits and libraries. Such kits provide a variety of functionalities, such as simulators, purpose-purpose programming languages, and many other resources to aid in the preparation of quantum code and its execution. The most well-known quantum software development kits are IBM Qiskit, Google Cirq SDK, and Microsoft Quantum Development Kit.

3.2.2. Quantum Software Libraries

Quantum Libraries consist of the code, functions and resources which are written in advance to make quantum algorithms and applications easier to create. These multipurpose hardware, programs and applications are engineered to exploit the massive potential of quantum computers [14]. Examples QuTip A good quantum optics and quantum information science package, which provides quantum system simulation and analysis tools. Cirq, another project developed by Google is also focused on quantum circuit design and noisy simulation but open fermion is focused on quantum chemistry simulation. The large technological companies and research centers are in the forefront of developing quantum software libraries. IBM, Google, Microsoft, and Rigetti are some of the well-known market players in this sector.

3.2.3. Quantum Cloud Services

Quantum cloud services are cloud-based services and tools that offer QC resources and tools. The services enable the users to execute quantum algorithms and programs without the need to own or operate their own quantum computers. The remote access to quantum processors is through cloud implementations by companies such as IBM, Google, and Righetti. IBM Quantum Experience is an access control to IBM quantum processors and simulators by offering cloud access, and Amazon Braket is a service offered by Amazon Web Services (AWS) that provides QC resources. On these platforms, it is feasible to test quantum algorithms, execute quantum simulations, and learn about the possibilities of QC. The simple model of quantum cloud computing software is presented in Figure 1. Advantages of quantum cloud services:

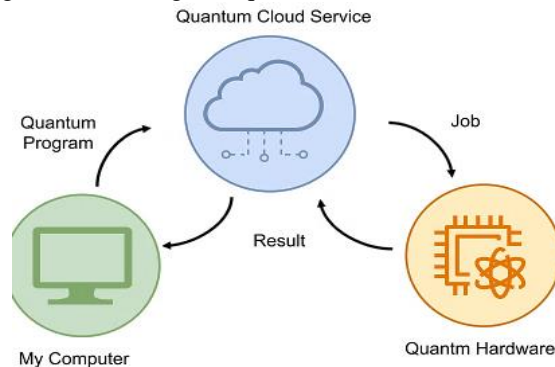


Fig 1: Model Of Quantum Cloud Computing

4. Agile Practices with Quantum Computing in Modern Engineering

Combination Agile and QC offers a successful, iterative process of quantum application development. The focus of Agile on short development cycles, cross-functional teams, and feedback can be easily adjusted to the experimental nature of QC, where the development progress is frequently incremental, and highly subject to technology change [15]. This implies that Agile offers a system to divide large and complex quantum algorithms into smaller, manageable tasks and also provides the teams with a system of responding to the technological and theoretical innovation in quantum. It operates with a constant improvement strategy on quantum solutions, which are optimized to serve business purposes and to serve machine learning and minimize risk and provide valuable changes.

4.1. Adapting Agile Practices to the Unique Nature of Quantum Software Engineering

The QSE should eventually incorporate the rapid development heritage of traditional computer programming, which started with hardwired, hardware-focused methodologies in the 1950s and matured into the present agile system development standards. However, the tools and methodologies used to accomplish agility within conventional computer science must be reassessed in light of the characteristics of quantum software development. Software developers meet unanticipated quantum laws and must adopt an entirely different programming paradigm while constructing quantum programs [16]. For instance, measuring the values of quantum instructions that are executed in the qubit state. Testing quantum programs entails addressing the following issues: (1) defining test oracles because a quantum program's state may be in superposition, making it difficult to determine its exact state; (2) effectively generate quantum test data since quantum variables can be exponentially higher than conventional variables; and (3) dealing with false positives or negatives produced by vulnerabilities or hardware failures.

4.2. Challenges and Solutions in Quantum Software Development

QC has many challenges to integrate into the software sector despite its potential. This is achieved through a fuzzy Analytic Hierarchy Process (AHP) technique that highlights issues such as skill deficiencies, high prices and technical limitations. The problems require the proactive role in addressing the skills gaps and building interrelations between the stakeholders of the field. a multivocal literature review of the challenges and solutions in QC, identification of the gaps in knowledge and requirements of new research. The authors in their work also highlight that to prevent such a scenario, the organizations must have an integrated framework which assists in recognizing how such challenges can be overcome [17]. The QC plays a significant part in the corporate aspect of the initiatives of project managers who desire to exploit the technology. Project managers capable of making effective decisions through understanding the cost and operation possibilities of QC that can be utilized in accordance with the company goals.

4.3. Agile Methodologies in Quantum Software Development

Quantum agile development finds its basis on the convergence of agile and QC. They attract attention in their paper to the versatility of agile concepts to quantum software development, and give an idea of the opportunities of applying QC by project teams to the development process of the project using the iterative development and the continuous improvement of the coherent changes in technological tools and advancements. talk about the digital revolution of project management, and the need to have flexible methodologies that can respond to unexpected changes in the technological tool and developments. Their work is in line

with the expanding understanding that the current principles of project management are to be modified according to the latest technologies, such as QC.

4.4. Impacts of Quantum Computing on Software Engineering

QC is transforming the field of software engineering to come up with new algorithms, complexity, and hybrid systems. It also becomes a stimulus to the subsequent progress of optimization and requires the transition to post-quantum cryptography.

- **Algorithm Development:** New schemes have been introduced through QC, such as the Grover algorithm to search unsorted databases as well as the Shor algorithm to factor large integers. Software developers may need to alter or create new algorithms on some tasks in order to be able to attain quantum speedup.
- **Increased Complexity:** Quantum computers are more complex and thus difficult to develop software. Software engineers are faced with special defects and performance concerns by quantum systems.
- **Hybrid Systems:** Several applications necessitate the integration of classical and QC. To properly leverage both types of processing, software developers must devise novel integration mechanisms.
- **Optimization Problems:** Finance and logistics are just some of the optimization challenges that QC can be successful at. Software engineers have to reconsider these challenges to be able to exploit the power of quantum algorithms.
- **Security and Cryptography:** Post-quantum cryptography has arisen in response to the challenge given by QC to traditional encryption techniques. To ensure security, software engineers must accept and implement the new requirements.

5. Literature Review

The following section offers a literature review concentrating on Understanding the Use of Agile Development Practices Alongside QC in Modern Engineering Projects, with a quick overview in Table I.

de Vicente Mohino et al. (2019) The suggested approach indicates that if security is considered from the start, vulnerabilities may be swiftly detected and repaired throughout the project's specified time, with no additional effort or expense to the client, boosting the chance of success in terms of both functionality and quality. The software development environment is meant to provide effective products in the shortest period of time and utilising the fewest resources feasible. In this context, critical components such as software quality or security are not even examined, and in most situations, the considerable value offered to projects is not taken into consideration. [18].

Pereira and Russo (2018) examined the relationship between agile software development approaches and the Design Thinking methodology. A thorough literature study was carried out to gather, analyse, and assess 29 articles on the topic. It shows that most integrated models may be applied across the software life cycle, with the most popular being Scrum, an agile methodology, and the ISO DT process. The software's quality and usability improved as a result of the integrated models' better congruence between the development team and end users [19].

Gustavsson (2016) provides a detailed literature study to establish the benefits of using agile approaches in non-software development activities. With a few exceptions pertaining to strategy formulation and library management, the most of the 21 case studies analysed were for industrial businesses. The most commonly mentioned advantages were teamwork, client communication, productivity, and flexibility. The primary advantages matched the Agile Manifesto's first value: persons and interactions vs processes and systems [20].

Matharu et al. (2015) Discuss the necessity, advantages, and rising popularity of Agile methodologies in light of current dynamic and challenging software needs. This study's main goal is to perform empirical research on the choice of three of the most popular agile approaches: Scrum, Extreme Programming, and Kanban. Additionally, this article analyses various techniques to agile software development. It illustrates that Scrum-based development is more commonly employed in the contemporary software industry than Extreme Programming and Kanban techniques [21].

Inayat et al. (2015) This research identified 17 agile requirements engineering options, five concerns with traditional demands engineering that agile demands engineering solves, and eight obstacles provided by agile requirements engineering. Their findings, however, imply that agile requirements engineering as a research framework warrants greater investigation. To fully comprehend the effects of agile requirements engineering techniques, such as managing non-functional demands and self-organising teams, more study is required [22].

Usman et al. (2014) Provide an in-depth overview of the present state of ASD effort estimation. The selection of twenty-five primary studies yielded the following significant findings The most typical size metrics in an agile context are use case points and story points; the most prevalent accuracy metrics are MMRE and MRE; team skills, prior experience, and task size are the three key cost drivers for effort estimate in ASD; The only two agile methodologies found in the major research are XP and SCRUM. [23].

Table I summarizes the literature on Understanding the Use of Agile Development Practices Alongside QC in Modern Engineering Projects, including methodology, significant findings, problems, and future directions

Table 1: Summary on Agile Development Practices alongside Quantum Computing in Modern Engineering Projects

Author	Study On	Approach	Key Findings	Challenges	Future Directions
de Vicente Mohino et al. (2019)	Integrating security early in Agile development	Proposed security-focused methodology within Agile projects	Early consideration of security enables timely detection and resolution of vulnerabilities; no added time or cost for clients; improves project success in terms of functionality and quality	Traditional Agile environments often overlook security and software quality due to focus on speed and minimal resources	Develop Agile frameworks where security is embedded throughout all iterations; explore automated security tools for early detection
Pereira & Russo (2018)	Integration of Design Thinking (DT) with Agile methodologies	Systematic Literature Review (29 studies)	Integrated DT–Agile models improve user alignment, software quality, and usability; ISO DT model + Scrum most commonly used	Limited empirical studies on combined DT–Agile models; inconsistent application across software life cycle	Conduct empirical evaluations of integrated DT–Agile processes; develop standard hybrid frameworks
Gustavsson (2016)	Agile methods' benefits in settings other than software development	Systematic Literature Review (21 case studies)	Agile encourages cooperation, customer involvement, productivity, and flexibility; the advantages are consistent with the Agile Manifesto ideal. "People and relationships over procedures and equipment"	Limited evidence from sectors beyond manufacturing; lack of long-term impact studies	Expand Agile research in diverse non-software domains; investigate scalability and sustainability of Agile practices
Matharu et al. (2015)	Adoption and comparison of Agile methodologies (Scrum, XP, Kanban)	Empirical study + comparative analysis	The most popular approach is Scrum, while Agile enhances adaptability to changing software requirements.	Lack of standardized criteria for selecting Agile methods; limited comparative performance data	Develop criteria for methodology selection; evaluate hybrid Agile models combining strengths of multiple methods
Inayat et al. (2015)	Practices and problems in ARE	Systematic Literature Review	Identified 17 ARE practices, five traditional RE challenges solved by Agile, and eight new challenges emerging from ARE	Challenges include handling non-functional requirements, limited documentation, self-organizing team issues	Conduct empirical studies on ARE effectiveness; explore strategies for managing non-functional requirements in Agile
Usman et al. (2014)	Effort estimation approaches in ASD	Systematic Literature Review (25 primary studies)	Subjective estimation dominates ASD; use-case points & story points extensively employed; MMRE & MRE routinely applied metrics; team skills, experience, and work size are significant cost factors; The most popular approaches are Scrum and XP.	Lack of objective estimation models; variability due to human factors in subjective estimation	Develop data-driven estimation models; examine estimation accuracy across different Agile frameworks

6. Conclusion and Future Work

Combining Agile development practices with quantum computing is likely to result in better engineering projects of the future. The eventuality that jewels cut by the adoption of the Agile methodology's iterative, collective, and adaptive characters is already here and it lies in the uncertainty and fast-paced changes surrounding the development of quantum technologies. New requirements that are accompanied with the latter include advanced model programming, faulty hardware and large test cases that can hardly be solved with standard software development methods. However, Scrum, Kanban, XP frameworks support the development teams to adjust to the changes which are currently happening by sharing the resources among the short sprints, undertaking constant experiments, and establishing regular feedbacks. The result of it is the more freedom is allowed; the less risk is engaged in the sooner quantum algorithms and applications will be perfected. Also, Agile practices are skill-gap fixer and that favours teamwork activity in functional areas and therefore, knowledge sharing. The QC is still at the infancy stage and the Agile techniques are right at the perfect period to make certain that the quantum solutions will not only imitate the business wants, but also the technical enhancement and engineering discoveries.

The future efforts should target professionalization of the existing Agile structures to quantum software engineering, quantum test equipment, and training courses to developers. Besides, it should be extended to hybrid classical-quantum workflows, automated debugging, and scaling up project management models of large-scale quantum applications exploration should be included in the research pipeline.

References

- [1] S. Gupta, N. Agrawal, and S. Gupta, "A Review on Search Engine Optimization: Basics," *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 5, pp. 381–390, May 2016, doi: 10.14257/ijhit.2016.9.5.32.
- [2] A. Rasnacic and S. Berzisa, "Method for Adaptation and Implementation of Agile Project Management Methodology," *Procedia Comput. Sci.*, vol. 104, pp. 43–50, 2017, doi: 10.1016/j.procs.2017.01.055.
- [3] V. Liubchenko, "A review of agile practices for project management," in 2016 XIth International Scientific and Technical Conference Computer Sciences and Information Technologies (CSIT), IEEE, Sep. 2016, pp. 168–170. doi: 10.1109/STC-CSIT.2016.7589897.
- [4] I. Ray, "Quantum Computing," 2011. doi: 10.13140/2.1.1021.7286.
- [5] C. G. Almudever et al., "The engineering challenges in quantum computing," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, IEEE, Mar. 2017, pp. 836–845. doi: 10.23919/DATE.2017.7927104.
- [6] V. R. Gotarane and S. S. M. Gandhi, "Quantum Computing: Future Computing," *Int. Res. J. Eng. Technol.*, vol. 03, no. 02, pp. 1424–1427, 2016.
- [7] A. M. M. Hamed and H. Abushama, "Popular agile approaches in software development: Review and analysis," in 2013 International Conference On Computing, Electrical And Electronic Engineering (Iccee), IEEE, Aug. 2013, pp. 160–166. doi: 10.1109/ICCEEE.2013.6633925.
- [8] S. Kumar and P. Dubey, "Software Development Life Cycle (Sdlc) Analytical Comparison and Survey on Traditional and Agile Methodology," *Natl. Mon. Ref. J. Res. Sci. Technol.*, vol. 2, no. 8, pp. 22–30, 2013.
- [9] M. Alqudah and R. Razali, "A comparison of scrum and Kanban for identifying their selection factors," *Proc. 2017 6th Int. Conf. Electr. Eng. Informatics Sustain. Soc. Through Digit. Innov. ICEEI 2017*, vol. 2017-Novem, no. April, pp. 1–6, 2017, doi: 10.1109/ICEEI.2017.8312434.
- [10] S. M. Saleh, M. A. Rahman, and K. A. Asgor, "Comparative Study on the Software Methodologies for Effective Software Development," *Int. J. Sci. Eng. Res.*, vol. 8, no. 4, pp. 1018–1025, 2017.
- [11] T. Forcer, T. Hey, D. Ross, and P. G. R. Smith, "Superposition, entanglement and quantum computation," *Quantum Inf. Comput.*, vol. 2, no. 2, pp. 97–116, Feb. 2002, doi: 10.26421/QIC2.2-1.
- [12] A. Acín et al., "The quantum technologies roadmap: A European community view," *New J. Phys.*, vol. 20, no. 8, 2018, doi: 10.1088/1367-2630/aad1ea.
- [13] R. LaRose, "Overview and comparison of gate level quantum software platforms," *Quantum*, vol. 3, 2019, doi: 10.22331/q-2019-03-25-130.
- [14] J. R. Johansson, P. D. Nation, and F. Nori, "QuTiP: An open-source Python framework for the dynamics of open quantum systems," *Comput. Phys. Commun.*, vol. 183, no. 8, pp. 1760–1772, 2012, doi: 10.1016/j.cpc.2012.02.021.
- [15] M. Kassab, J. DeFranco, and V. Graciano Neto, "An Empirical Investigation on the Satisfaction Levels with the Requirements Engineering Practices: Agile vs. Waterfall," in 2018 IEEE International Professional Communication Conference (ProComm), IEEE, Jul. 2018, pp. 118–124. doi: 10.1109/ProComm.2018.00033.
- [16] S. Rajagopalan and S. K. Mathew, "Choice of Agile Methodologies in Software Development: A Vendor Perspective," *J. Int. Technol. Inf. Manag.*, vol. 25, no. 1, 2016, doi: 10.58729/1941-6679.1251.
- [17] S. Mnqonywa and S. von Solms, "A systematic literature review of the agile methodology applied during construction project design," *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, vol. 2018, no. NOV, pp. 1430–1442, 2018.
- [18] J. de Vicente Mohino, J. Bermejo Higuera, J. R. Bermejo Higuera, and J. A. Sicilia Montalvo, "The Application of a New Secure Software Development Life Cycle (S-SDLC) with Agile Methodologies," *Electronics*, vol. 8, no. 11, p. 1218, Oct. 2019, doi: 10.3390/electronics8111218.
- [19] J. C. Pereira and R. de F. S. M. Russo, "Design Thinking Integrated in Agile Software Development: A Systematic

- Literature Review,” *Procedia Comput. Sci.*, vol. 138, pp. 775–782, 2018, doi: 10.1016/j.procs.2018.10.101.
- [20] T. Gustavsson, “Benefits Of Agile Project Management In A Non-Software Development Context A Literature Review,” *Proj. Manag. Dev. – Pract. Perspect.*, vol. 5, no. 8, pp. 67–82, 2016.
- [21] Polu, A. R., Buddula, D. V. K. R., Narra, B., Gupta, A., Vattikonda, N., & Patchipulusu, H. (2021). Evolution of AI in Software Development and Cybersecurity: Unifying Automation, Innovation, and Protection in the Digital Age. Available at SSRN 5266517.
- [22] Padur, S. K. R. (2020). From centralized control to democratized insights: Migrating enterprise reporting from IBM Cognos to Microsoft Power BI. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, 6(1), 218-225.
- [23] Bitkuri, V., Kendyala, R., Kurma, J., Mamidala, V., Enokkaren, S. J., & Attipalli, A. (2021). Systematic Review of Artificial Intelligence Techniques for Enhancing Financial Reporting and Regulatory Compliance. *International Journal of Emerging Trends in Computer Science and Information Technology*, 2(4), 73-80.
- [24] Padur, S. K. R. (2019). Machine learning for predictive capacity planning: Evolution from analytical modeling to autonomous infrastructure. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 5(5), 285-293.
- [25] Attipalli, A., Enokkaren, S., BITKURI, V., Kendyala, R., KURMA, J., & Mamidala, J. V. (2021). Enhancing Cloud Infrastructure Security Through AI-Powered Big Data Anomaly Detection. Available at SSRN 5741305.
- [26] Singh, A. A. S., Tamilmani, V., Maniar, V., Kothamaram, R. R., Rajendran, D., & Namburi, V. D. (2021). Predictive Modeling for Classification of SMS Spam Using NLP and ML Techniques. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 2(4), 60-69.
- [27] Padur, S. K. R. (2020). AI augmented disaster recovery simulations: From chaos engineering to autonomous resilience orchestration. *International Journal of Scientific Research in Science, Engineering and Technology*, 7(6), 367-378.
- [28] Reddy Padur, S. K. (2021). From Scripts to Platforms-as-Code: The Role of Terraform and Ansible in Declarative Infrastructure Rollouts. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 621-628.
- [29] Kothamaram, R. R., Rajendran, D., Namburi, V. D., Singh, A. A. S., Tamilmani, V., & Maniar, V. (2021). A Survey of Adoption Challenges and Barriers in Implementing Digital Payroll Management Systems in Across Organizations. *International Journal of Emerging Research in Engineering and Technology*, 2(2), 64-72.
- [30] Padur, S. K. R. (2018). Autonomous cloud economics: AI driven right sizing and cost optimization in hybrid infrastructures. *International Journal of Scientific Research in Science and Technology*, 4(5), 2090-2097.
- [31] Rajendran, D., Namburi, V. D., Singh, A. A. S., Tamilmani, V., Maniar, V., & Kothamaram, R. R. (2021). Anomaly Identification in IoT-Networks Using Artificial Intelligence-Based Data-Driven Techniques in Cloud Environmen. *International Journal of Emerging Trends in Computer Science and Information Technology*, 2(2), 83-91.
- [32] Padur, S. K. R. (2021). Bridging Human, System, and Cloud Integration through RESTful Automation and Governance. *the International Journal of Science, Engineering and Technology*, 9(6).
- [33] Attipalli, A., BITKURI, V., KURMA, J., Enokkaren, S., Kendyala, R., & Mamidala, J. V. (2021). A Survey of Artificial Intelligence Methods in Liquidity Risk Management: Challenges and Future Directions. Available at SSRN 5741342.
- [34] Padur, S. K. R. (2021). From Control to Code: Governance Models for Multi-Cloud ERP Modernization. *International Journal of Scientific Research & Engineering Trends*, 7(3).
- [35] Routhu, K. K. (2021). Harnessing AI Dashboards in Oracle Cloud HCM: Advancing Predictive Workforce Intelligence and Managerial Agility. *International Journal of Scientific Research & Engineering Trends*, 7(6).
- [36] Padur, S. K. R. (2021). Deep learning and process mining for ERP anomaly detection: Toward predictive and self-monitoring enterprise platforms. Available at SSRN 5605531.