

New Hybrid Model with Fine Blend of Agile and Waterfall

P Vijaya Vardhan Reddy
SM-IEEE,
IEEE, India.
drvijayvardhanreddy@ieee.org
<https://www.drvijayvardhan.com>

Abstract

The paper recommends a new hybrid model for software projects delivery with fine blend of the both agile and waterfall models by addressing the drawbacks of both models. The traditional waterfall model development goes one phase after the other and any changes to previous phase is addressed late in the delivery cycle thus most of the times implemented software and customer expectation will not be in sync or sometimes poles apart. The trending agile focuses on small chunks development with iterative cycles, going back to the customer for feedback at every phase and reaching the expectation and delivering the functionality which client was looking for. The agile model may not be suitable for projects where the requirements are very intricate and cannot be easily fragmented into smaller iterations and requirements are definite and constant, and where deviations to the requirements are improbable. The waterfall model may not be suitable for projects where the requirements are uncertain or likely to change, as changes to the requirements need changes to previous phases. The paper recommends a new fine blended model with the best of agile and waterfall approaches with the collaboration of Delivery with Sprints and POD teams, and gets the delivery with optimal cost and speed. The projected novel model is a fine blend of best of both models by eliminating shortfalls of the Agile and Waterfall.

Keywords: Software Project; Agile; Waterfall; POD; Sprint; Delivery.

1. Introduction

The paper proposes a novel delivery model for a software engineering project with the combination of both agile and waterfall models. Further in the model, the implementation and delivery carried out with the Epics divided into multiple Sprints with work spread as iterations and as well sequential depending on the phase of the project. The multiple sprints development with POD (Distinguished Talented Resources) teams goes in iterative to get feedback from customer and to speed up the delivery. The phases like User Acceptance Testing (UAT) and Quality Assurance (QA) bug fixing, from Development to System Testing and Quality Assurance and User Acceptance Testing goes in sequential to make sure that by fixing one bug another bug is not introduced and code repetition is less and common services utilization is more. Based on the necessity and requirements, the software development phases will go sequential and iterative to provide cost effective delivery with apt time to market.

The proposed novel model in the paper uses best of both the models with iterative and sequential approaches based on the project requirements. If the requirement is complex in nature and not able to split into smaller workable components that phase of development goes in sequential as adapted from the waterfall delivery model. In the phases where requirements are unclear and then the development of those phases will adapt an iterative process as per an agile delivery model. The software delivery phases like requirements gathering and development phases will run parallel to expedite the delivery and time to market as opposed to the waterfall model. The bug fixing and testing phases goes in sequential way to keep robustness in the system without more bugs and no rework of development and quality. As an output, a fine blend of the both models are adapted and a new model is proposed to the projects to remove the ambiguity, clear work-breakdown structure, Gantt Chart in a phased manner to have the benefits of delivery speed and early time to market.

The software project implementation in complete agile model takes shorter duration and less cost and time to market delivery is quite possible. But certain complex tasks in (safety-critical systems) real time applications cannot be broken down in to smaller pieces hence in the proposed blended model recommends complex applications in waterfall model to go in sequential way Everything in agile introduces repetitive code, and less robustness of the system. Hence, quality tasks in sequential way to deliver robust system with bug free and no respective code along with important factor of reducing Technical Debt.

2. Related Work

The current trending agile and traditional waterfall delivery models are most extensively used models in the delivery of any software engineering projects. The waterfall model referred to as linear-sequential life cycle model. Various problems have been reported related to the model. Commonly accepted problems are for example to cope with change and that defects all too often are detected too late in the software development process [Luettavissa (2015)]. The waterfall is a sequential design process, often used in software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, implantation, testing, production, and maintenance [Kramer, Mitch (2004)].

The waterfall model Problems are, that the model does not cope well with change, generates a lot of reworks, and leads to unpredictable software quality due to late testing [Jardim *et al.* (2015)]. Despite the problems identified, the model is still widely used in software industry, some researchers are even convinced that it will be around for a much longer period [Luettavissa (2015)]. The waterfall model software development life cycle is perceived as the timeframe that spans from the development of a new system to its eventual retirement. It is a process that starts with the emergence of an idea, goes through its implementation, and ends with its termination, moving across all the intermediate stages in which its viability and usability are prioritized [Luettavissa (2015)].

The shortcoming of waterfall model is that it does not allow much reflection or revision. working software is not produced until late during the life cycle. Scope adjustment during the life cycle can end a project. High risk and uncertainty. It is difficult to measure progress at every stage of the model and the time, cost is not determined. Moreover, the integration is done at the end which does not even explain the identification of challenges and business bottlenecks in business life cycle. As the steps are interrelated and dependent on each other a lot of time is wasted. Coordination is very important which is not possible in simple method like waterfall [Jardim *et al.* (2015)].

The Agile and Waterfall models both offers best benefits and issues in project characteristics [Balaji *et al.* (2007)]. Agile techniques vary in practices and emphasis, they share common characteristics, including iterative development and a focus on interaction, communication, and the reduction of resource-intensive intermediate artifacts. Furthermore, Agile approaches combine short iterative cycles with feature planning and dynamic prioritization [Sharma *et al.* (2012)]. As explained that each iteration in an iterative product, is a self-contained, mini-project with activities that span requirements analysis, design, implementation, and test [Srivastava *et al.* (2017)].

It seems that the informal evaluation techniques of agile processes may not be sufficient for establishing the quality of safety-critical systems [McCormick, Mike (2012)]. In Agile it is difficult to see frequent releases in short development cycles, pair programming, starting development before completing requirements and an ever-evolving project direction applicable to large-scale software projects [Abrahamsson *et al.* (2016)]. The coordination of a higher number of projects and tests on system levels, the complex decision processes, the creation of baselines for a high number of internal releases and the management overhead all form obstacles for large-scale, multi-team Agile development [Oueslati *et al.* (2010)]. The ideas on which Agile or plan-driven methodologies are popular, how methodologies are perceived by developers, how methods could be combined [Adamson *et al.* (2010)]. A framework for deciding whether a project should be developed Agile or in a Waterfall Model or through a model somewhere in between is important [Balaji *et al.* (2007)]. It is important to think to balance Agile and plan-driven methods [Boehm *et al.* (2005)]. Lot of paradigm shift is required to move from waterfall to agile in traditional organizations [Stretcu *et al.* (2023)]. The model proposed specific number of sprints [Reddy, P.V.V. *et al.* (2023)] hence resulted solution which is of a specific in nature. To avoid this, I have chosen multiple number of sprints and proposed an effective generic solution to adapt. Adapting Agile processes to a distributed environment in large organizations is key task [Tudor *et al.* (2006)]. Most of the organizations now finding the benefits of agile and moving towards it but caution is required [Strode *et al.* (2009)]. As lot of IT projects now being developed in agile than mission critical systems paving way for agile [Malik *et al.* (2019) and Kumar *et al.* (2012)]. However, I recommend in this paper a balanced approach need to be adapted with fine blend of both models based on phase wise with benefits of optimal cost and speed.

3. New Hybrid Model with Fine Blend of Agile and Waterfall

The proposed new delivery model is the fine blend of both the delivery models, the agile and the waterfall. The proposed novel model removes the bottlenecks of the agile delivery model with sequential flow and removes the drawbacks of the waterfall model with multiple small iterations to have the reality check on the progress accuracy. The proposed model recommends multiple POD structure for faster and parallel implementation of software engineering projects to give delivery to the customer and time to market in with optimal cost and speed. The quality and user acceptance test reported bug fixing goes for development to system testing to quality assurance to user acceptance in sequential way to avoid re-introduction of bugs, repetitive code even there is a little bit delay instead of going these phases in complete agile way. As too much agile is also not required which causes significant side effects on the system. Hence, as depicted in Fig 1., our proposed novel model is fine blend of agile and waterfall. First half the in Fig 1., goes in agile way to achieve optimal cost and speed and second half Dev to UAT goes in sequential way to achieve robustness in the system.

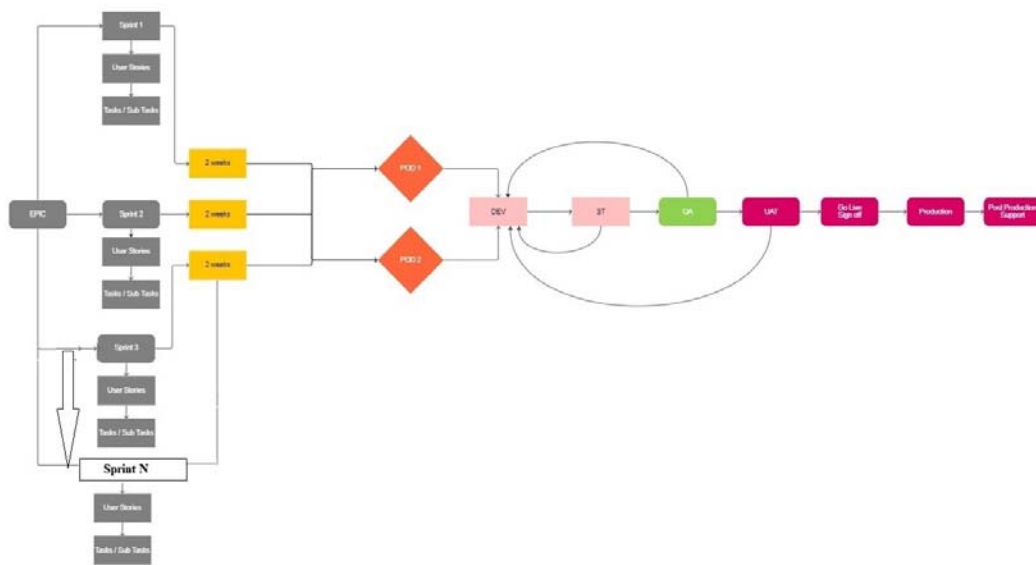


Fig. 1. New Hybrid Model with Fine Blend of Agile and Waterfall

The novel delivery model recommends the concept dividing the Epics into Sprints and Sprints into User Stories and further drill-down as User Stories to Tasks and Sub-Tasks as per Lean Agile Methodology. The duration of each sprint is suggested as 2 weeks iterative cycles to have faster time to market. The Sprints are implemented with multiple PODs with different competences. Now in each Sprint which is developed by a POD, the Development (Dev), System Testing (ST), Quality Analysis (QA), and User Acceptance Testing (UAT) will be run sequentially with the adaptation of waterfall. Any bugs reported in UAT comes back to Dev, and the sequential process of Dev to ST to QA to UAT continues to deliver the project with utmost care of quality control with one more rigorous all sequential test phases as illustrated in Fig. 1. After UAT, Go-Live Sign Off, move to Production and post Production activities take place one after the other sequentially.

The new delivery model used the division of Epics, Sprints, and PODs in an Agile way with iterative cycles and the development process of Dev, ST, QA, UAT sequentially in a waterfall way to give the best results. Even post UAT, if any bugs are reported, the proposed model recommended again the sequential flow of Dev, ST, QA, UAT to deliver the robust system with high quality project. The fine blend of the both agile and waterfall models gives the utmost acceptable project deliverables by the customer which is the USP of the proposed new delivery model.

4. Blended WBS for Software Engineering Project

The Work-Breakdown structure (WBS) is proposed with the blend of agile and waterfall delivery models. The WBS, splits the software engineering project into Epics, Story, Tasks, and Sub-Tasks.

Blended WBS (Agile and Waterfall) for Software Project	Resources	Effort (PDs)
Epic	Team	1
Software Engineering project implementation (Discovery of Technologies)	Architect	3
Story		
Create a documentation of Project Modules	BA	2
The modules should be able to give complete functionality	BA	2
Tasks / Sub-Tasks		
Document Study	Developer	4
Setup environment for project services development	Developer	4
Writing code for modules implementation	Developer	5
The modules should be able to give the solution to proposed requirements	Developer	2
Implement the modules for the project	Developer	2
Complete the integration with the help of DevOps Team	Developer	3
System Testing and Load Testing	QA	3
Go - Live	Team	1
	Total	32

Table 1. Blended WBS for Software Engineering Project

As depicted in the Table 1, Work-Breakdown structure is constructed with an Agile delivery model of Epics, Story, Tasks, and Sub-Tasks. The POD resources are allocated to the software engineering project. The effort estimations and dependencies are given in a sequential flow as per Waterfall delivery model. Total estimation of 32 PDs (Person Days) for development of a project if it proceeds in complete waterfall model way.

5. Gantt Chart Timelines with Waterfall Model

The Gantt Chart is presented below with completion timelines with complete waterfall delivery method. The Gantt Chart for the software engineering project proposes set of tasks and timelines with waterfall model, as depicted in Table 2. If the project is implemented in complete waterfall model way, it takes 32 days to complete as illustrated in Gantt Chart with completion timelines.

Development Plan (Software Engineering Project)	June / July																
Overall Timeline (Dates)	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Initiation	█																
Execution		█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Software Engineering Project Implementation	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Discovery (Technologies)		█	█	█	█	█											
BRD / FSD			█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Development				█	█	█	█	█	█	█	█	█	█	█	█	█	█
SIT							█	█	█	█	█	█	█	█	█	█	█
LT							█	█	█	█	█	█	█	█	█	█	█
EUT																	
Pre-Go Live (Sign Off)																	█
Go Live																	█
Post Go Live	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Support																	█

Table 3. Gantt Chart Timelines with Agile Model

All software phases from imitation to Go-Live i.e., discovery, Business Requirements, Functional Specifications, Development, System testing, Load testing, QA, End User testing and Go-Live goes in agile way with iterative cycles, it takes 13 days to complete the project as depicted in Gantt Chart with completion timelines.

The software project implementation in complete agile model takes shorter duration and less cost and time to market delivery is quite possible. Reaching customer expectations are also guaranteed at the end of the delivery. But on a cautious note, everything in agile likely to introduces repetitive code, high technical debt, re-introduction of closed bugs, and less robustness of the system.

7. Gantt Chart Timelines with Fine Blend of Agile and Waterfall

The Gantt Chart is presented below with completion timelines with fine blend of an Agile and the Waterfall delivery models. The Gantt Chart for the software engineering project proposes set of tasks and timelines with both agile and waterfall models based on the necessity with fine blend, as depicted in Table 4. If the project is implemented in fine blend of agile and waterfall model way, it takes 16 days to complete as illustrated in Gantt Chart with completion timelines.

Development Plan (Software Engineering Project)	June / July																					
	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Overall Timeline (Dates)																						
Initiation																						
Execution																						
Software Engineering Project Implementation																						
Discovery (Technologies)																						
BRD / FSD																						
Development																						
SIT																						
LT																						
EUT																						
Pre-Go Live (Sign Off)																						
Go Live																						
Post Go Live																						
Support																						

Table 4. Gantt Chart Timelines with Fine Blend of Agile and Waterfall

As illustrated in the Table 4, the blended Gantt Chart is represented with blend of an agile delivery model of parallel implementation and sequential implementation of the waterfall model. The Business Requirements Documentation (BRD), the Functional Specification Document (FSD) starts a day head of development activities and runs in parallel with development activity for about 3 days. (The red blocks indicate the weekend holidays). In the same way, as per agile delivery model, System Integration Testing (SIT), and Load Testing (LT) runs in parallel with development tasks for about 3 days. As per proposed blended delivery model, the Discovery task starts a head of BRD/FSD and the discovery activities runs sequential for 3 days. The complex discovery task activities run completely sequential for 3 days as per the water fall model. Hence, as per the proposed novel delivery model, the complex discovery section runs in sequentially as a sole activity in silos as per the waterfall model the development activity runs in parallel with BRD/FSD activity and in parallel with SIT/LT activity as per the agile model. The proposed blended delivery model recommends the activities in both the agile and the waterfall delivery model to achieve the best outcome and removes the drawbacks of the both models.

The drawback of an agile delivery model is where certain complex activities cannot be broken down into smaller iterations, there required a sequential flow of activities as per the waterfall model. The proposed novel model in this paper, as illustrated in the Gantt Chart, the Discovery section is completely sequential activity to address the complexity where the agile delivery model is not an appropriate choice. The development activities as depicted in Gantt Chart runs sequentially for certain period of 3 days where those activities cannot be broken down into smaller iterations hence recommended to run sequentially as per the waterfall model where an agile delivery model is not suitable. The bug fixing phase from Dev to UAT goes in sequential way to create robust system without introducing more bugs while fixing the other bugs and no repetitive code. The drawback of a waterfall delivery model is where requirements are not clear and can run only sequentially taking longer time to market. In the proposed novel model, the waterfall delivery model drawbacks are addressed. As per Gantt Chart, where BRD/FSD is all about gathering the requirements, hence these activities run in parallel with development activity in iterative way to accommodate dynamic changes as per an agile delivery model where the waterfall model is not an apt choice. As to address time to market criteria the novel model recommends SIT/LT activities runs in parallel with development activities. The proposed model addresses drawbacks of the both agile and

waterfall delivery models and recommends the novel delivery model which enhances the productivity, clears the ambiguity, and speeds up delivery to give the advantage of time to market criteria.

Models	Effort for Completion (PDs)
Only Waterfall	32
Only Agile	13
Blend of Both Agile and Waterfall	16

Table 5. Summary of Efforts for Individual and Blended

The Waterfall model takes effort duration of 32 days and Agile model takes 13 days duration as illustrated in Table 5. The proposed fine blended model takes 16 days duration but with advantages over both agile and waterfall where individually agile or waterfall cannot give.

The software project implementation in complete waterfall model takes longer duration and higher cost and time to market delivery is not possible. Reaching customer expectations at the end of the delivery with probable deviations are also not guaranteed. This drawback addressed in proposed fine blended model where requirements gathering, development (along with system testing, load testing phases) go in parallel way and this iterative way guarantees the customer expectation and speed of the delivery in 16 days as compared to 32 days of waterfall model.

The software project implementation in complete agile model takes shorter duration and less cost and time to market delivery is quite possible. But certain complex tasks like discovery section cannot be broken down in to smaller pieces hence in the proposed blended model recommended discovery section in waterfall model to go in sequential way even though duration takes 3 days longer. Everything in agile introduces repetitive code, tech debt, re-induction of bugs, and less robustness of the system. Hence, UAT to Dev complete cycle Dev-ST-QA-UAT in sequential way to deliver robust system with bug free and no repetitive code along with important factor of reducing Technical Debt. As per proposed fine blended hybrid model, the effort duration is 16 days which is less compared to waterfall model with customer expectation and speed, optimal cost is guaranteed, and which is slightly high to agile model delivery of 13 days but system robustness, no repetitive code and Technical Debt reduction is achieved by going required and necessary phases in sequential way which is very important factor in today's world of everything agile way which is increasing technical debt unknowingly which is aptly curtailed with novel fine blended model. Hence, the proposed fine blended hybrid model achieves all the factors, customer expectations, speed (time to market), optimal cost, robust system with less technical debt.

8. Conclusion

The paper recommends the hybrid delivery model with fine blend of Agile and Waterfall delivery models with an example of software engineering project implementation.

An agile delivery model limitations are where complex activities cannot be broken down into smaller iterations. The waterfall delivery model limitations are when requirements are not clear changes in all phases are required with more rework and can run only sequentially, and takes longer time to deliver the project.

The proposed novel model in the paper addresses the drawbacks of both the models and presents the fine blend based on the requirements and suitability with which the project is delivered in time with required quality. As per the proposed model in this paper, as demonstrated in the Gantt Chart, the Discovery section is completely sequential to address the complexity. The discovery activities as represented in Gantt Chart runs sequentially because those activities cannot be broken down into smaller iterations. Hence, the proposed model recommends discovery complex activities to run sequentially. In the proposed novel model, as per Gantt Chart, where BRD/FSD is completely about gathering the requirements, hence these activities run in parallel with the development activity in an iterative way to accommodate dynamic changes which address the limitations of the waterfall model.

The software project implementation in complete waterfall model takes longer duration and higher cost and time to market delivery is not possible. Reaching customer expectations, the at end of the delivery are also not guaranteed with probable deviations. This drawback addressed in proposed fine blended model where requirements gathering, development (along with system testing, load testing phases) go in parallel way and this iterative way guarantees the customer expectation with timely feedback form customer and speeds up the delivery compared to the delivery time of waterfall model with no deviation from customer expectations.

The software project implementation in complete agile model takes shorter duration and less cost and time to market delivery is quite possible. But certain complex tasks like discovery section cannot be broken down in to smaller pieces hence in the proposed blended model recommended discovery section in waterfall model to go in sequential way even though duration takes 3 days longer. Everything in agile introduces repetitive code, tech debt, reintroduction of bugs, and less robustness of the system. Hence, UAT to Dev complete cycle Dev-ST-QA-UAT in sequential way to deliver robust system with bug free and no repetitive code along with important factor of reducing Technical Debt.

As per proposed fine blended new hybrid model, the effort duration is less compared to waterfall model with customer expectation and speed, optimal cost is guaranteed, and effort duration is slightly high of agile model delivery but system robustness, no repetitive code and Technical Debt reduction is achieved by going required and necessary phases in sequential way which is very important factor in today's world of everything agile, which is increasing technical debt unintentionally and is rightly curtailed with novel fine blended model. Hence, the proposed fine blended hybrid model achieves all the required important factors including customer expectations, speed (time to market), optimal cost, robust system with less technical debt.

9. Conflicts of interest

The authors have no conflicts of interest to declare.

10. Acknowledgments

No Funding was required or made for the proposed research.

References

- [1] Model, Waterfall. "Waterfall model." Luettavissa: <http://www.waterfall-model.com/>. Luettu 3 (2015).
- [2] Kramer, Mitch. "Best practices in systems development lifecycle: An analysis based on the waterfall model." *Review of Business & Finance Studies* 9, no. 1 (2018): 77-84. Sommerville, I.: Software Engineering, 7th edition. Pearson Education Ltd., London (2004).
- [3] Jardim, R. R. J., Marcos Santos, E. C. D. O. Neto, E. da Silva, and F. C. M. M. De Barros. "Integration of the waterfall model with ISO/IEC/IEEE 29148: 2018 for the development of military defense system." *IEEE Latin America Transactions* 18, no. 12 (2020): 2096-2103. Pedro Isaías and Tomayess Issa "High Level Models and Methodologies for Information Systems" ISBN 978-1-4614-9253-5 Springer (2015).
- [4] Sharma, Sheetal, Darothi Sarkar, and Divya Gupta. "Agile processes and methodologies: A conceptual study." *International journal on computer science and Engineering* 4, no. 5 (2012): 892.
- [5] Srivastava, Apoorva, Sukriti Bhardwaj, and Shipra Saraswat. "SCRUM model for agile methodology." In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 864-869. IEEE, 2017.
- [6] McCormick, Mike. "Waterfall vs. Agile methodology." *MPCS, N/A* 3 (2012).
- [7] Balaji, Sundramoorthy, and M. Sundararajan Murugaiyan. "Waterfall vs. V-Model vs. Agile: A comparative study on SDLC." *International Journal of Information Technology and Business Management* 2, no. 1 (2012): 26-30. Laurie Williams. A survey of agile development methodologies, 2007.
- [8] Abrahamsson, Pekka, Antti Hanhineva, Hanna Hulkko, Tuomas Ihme, Juho Jääliñoja, Mikko Korkala, Juha Koskela, Pekka Kyllönen, and Outi Salo. "Mobile-D: an agile approach for mobile application development." In *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, pp. 174-175. 2004. Bhuvan Unhelkar. *The Art of Agile Practice: A Composite Approach for Projects and Organizations*. CRC Press, 2016
- [9] Oueslati, Hela, Mohammad Masudur Rahman, and Lotfi ben Othmane. "Literature review of the challenges of developing secure software using the agile approach." In *2015 10th International Conference on Availability, Reliability and Security*, pp. 540-547. IEEE, 2015. Dave West, Tom Grant, M Gerush, and D D'silva. Agile development: Mainstream adoption has changed agility. *Forrester Research*, 2(1):41, 2010.
- [10] Adamson, David, Gregory Dyke, Hyeju Jang, and Carolyn Penstein Rosé. "Towards an agile approach to adapting dynamic collaboration support to student needs." *International Journal of Artificial Intelligence in Education* 24 (2014): 92-124. Cynick Young and Hiroki Terashima. How did we adapt agile processes to our distributed development? In *Agile, 2008. AGILE'08. Conference*, pages 304-309. IEEE, 2008
- [11] Boehm, Barry, and Richard Turner. "Management challenges to implementing agile processes in traditional development organizations." *IEEE software* 22, no. 5 (2005): 30-39.

- [12] Stretcu, Otilia, Edward Vendrow, Kenji Hata, Krishnamurthy Viswanathan, Vittorio Ferrari, Sasan Tavakkol, Wenlei Zhou et al. "Agile modeling: From concept to classifier in minutes." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22323-22334. 2023.
- [13] Reddy, P.V.V. (2023). Novel Delivery Model with the Combination of Iterative and Sequential Models. In: Arai, K. (eds) *Proceedings of the Future Technologies Conference (FTC) 2023, Volume 2*. FTC 2023. Lecture Notes in Networks and Systems, vol 814. Springer, Cham. https://doi.org/10.1007/978-3-031-47451-4_35
- [14] Strode, Diane E., Sid L. Huff, and Alexei Tretiakov. "The impact of organizational culture on agile method use." In *2009 42nd Hawaii International Conference on System Sciences*, pp. 1-9. IEEE, 2009.
- [15] Malik, Rabia Saeed, Sayed Sayeed Ahmad, and Muhammad Tuaha Hammad Hussain. "A review of agile methodology in IT projects." In *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*. 2019.
- [16] Kumar, Gaurav, and Pradeep Kumar Bhatia. "Impact of agile methodology on software development process." *International Journal of Computer Technology and Electronics Engineering (IJCTEE)* 2, no. 4 (2012): 46-50.
- [17] Tudor, Dot, and George A. Walter. "Using an agile approach in a large, traditional organization." In *AGILE 2006 (AGILE'06)*, pp. 7-pp. IEEE, 2006.

Authors Profile



Dr P Vijaya Vardhan Reddy is seasoned IT Professional and Research Academician with 23+ Years experience in IT industry and Research. He had awarded Doctorate in Computer Science and Engineering. He worked as a CTO @Ascent, VP @Machint, PM @Rify, GM-IT @HRC, AVP @GE, SSA @Saytam and SMT @ADP. He published a patent with Indian Patent Office (IPO). He has 13 research publications as first author in various international journals and conferences like Elsevier, Springer, and IEEE. There are 107 citations of his published research work. He received 27 peer review invitations to judge other authors' work. He is a certified Program Management Professional (PgMP), Project Management Professional (PMP) by PMI USA, a certified SAFe Agilist by Scaled Agile USA, and a CSM by Scrum Alliance USA.