International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

June 2021 | Issue 47 | Volume 13 | Number 2 | Pages 7-13

ARQI: MODEL FOR DEVELOPING WEB APPLICATION

A. Kalwar [1] R. Ajmera [2] A. Gill [3]

1. Department of Computer Science and Engineering, JECRC University, Jaipur, India, anju.kalwar@gmail.com
2. Department of Information Technology, JECRC University, Jaipur, India, reema.ajmera@jecrcu.edu.in
3. Department of Electrical Engineering, JECRC University, Jaipur, India, aamangill.87@gmail.com

Abstract-Web applications are immensely important in today's world. Every business whether small or big is focusing on web-based development. However, a significant number of web application projects fail each year. Some web application development project fails in pre-launch stage as their development becomes increasing complex and unmanageable. The reason for this is that web applications are developed using the lifecycle models designed for software development. Web applications have fundamentally different and unique characteristics as compared to software development such as short go to market time and global audience. The aim of this research is to propose a new iterative lifecycle model for web application development. The proposed model is designed to specifically cater to the development needs of web applications and the developed model is tested by practical implementation in case study. The results reveal that the proposed model leads better Business Value Management and consumer satisfaction.

Keywords: ARQI, EVM, Technical debt, Work capacity.

1. INTRODUCTION

The paradigm of development has shifted from desktop rich applications to web based applications. This has made internet a plethora of multi domain web applications that harness the power of internet-based technologies for accomplishing various objectives such as legacy systems, information systems, databases, workflow management, distributed knowledge, cooperative work, media sharing and many others. Web applications have now become a dominant and highly attractive platform for deploying business and are extensively used in information delivery platforms, social networking sites, e learning environments, ecommerce systems and many others.

The development of web applications may seem analogues to conventional software development. But web applications have different features and unique characteristics that define them different from classic software systems.

Most of the web applications are not carefully planned and face security breaches, site crash, server overload and performance failures. This scenario leads to loss of customers and consequently loss in revenue, devaluation of stocks and blemished reputation. In a more aggravate situation the company may face permanent loss of customers as well as lawsuits [1]. The legal implications levied on a company if the web application fails can be found in [2]. A Massachusetts based, Information Research Company, Cutter Consortium, conducted a survey to identify the problems faced during development of web application.

The development of web applications is a complex procedure. A web application development procedure must ensure integrity, quality and performance of the system. There are many hurdles in the development process such as flawed design, short sighted goals, lack of vision etc. Most of the present development methodology are designed according to the need of software engineering and are unable to capture the development and maintenance requirement of web application development.

Therefore, traditional methodologies of developing software applications cannot ensure quality development of web applications as well**.** There exists immediate need to develop an iterative development life cycle model which can space specifically cater to the dynamic changing needs of where application development and can ensure quality in each development phase.

The problem of increasing outputs quality is topical problem. Quality improvement process combines the activities of many multi-disciplines and is required not only for profit, and most importantly for the community as a whole and its interests [3].

The problem can be summarized as "need of an iterative development life cycle model that is easy to control and can simplify the development of web application by proper requirement gathering and limiting number of iterations". It is highly important in management of resources like employee forces, financial options, technological and technical facilities, or any other subjects contenting engineering executions [4].

2. LITERATURE SURVEY

Common request broker architecture (CORBA) was the first methodology proposed for the development of web applications in 1997 [5]. Web composition was another object-oriented technique proposed for the development of web application in 1999 [6]. The object-oriented approach for web application development considered only abstract features of a web application such as specific web links and navigation but did not include entire development process of web application.

The next era witnessed UML based approaches for web application development [7]. web application development is linear sequential flow in both approach object-oriented approach and UML based approach for and unable to include the need of continuous evolution with changing customers need and Technology, during the life cycle of web application development. In 2001 a family of processes called Agile Manifesto was drafted, which defined the development process for software engineering. [8] presented another approach called Extreme programming (XP) for developing web applications. This approach was based on Agile methodology. Agile methodology for web development gained popularity in 2008 [9]. The concept of agile methodology for web application emphasizes that technical Advancement prototypes must be released iteratively as per the customers need and during the life cycle of the web application development.

[10] counteracted the problems of Model Driven Development forward engineering when manual and automatic code updates occur in parallel and suggested. An online tool for rapid prototyping of web applications is demonstrated. The tool develops prototype of the web application and helps in studying the user interaction with the web application. In other words, different variations of an application can be developed in a short time period by rapidly modifying the application model and generating realistic prototypes, easily turned into deployable applications. This tool doesn't support requirement election, design standard. It supports only complexity management and attention of design.

[11] productivity is increased in web application development to model the system by reducing the effort, within system requirements the consistency is maintained by reducing the inconsistencies in requirement gathering stage. In the proposed framework a detailed modeling of requirements is performed before design phase.

[12] Feature tree named tool is developed to manage requirements and their changes that too helps in managing the reequipment changes and keeps the track of project status in two phases that helps the project manager.

Lack of documentation, poor architectural structure and less focus on design are its major drawbacks that affects the performance of extreme programming and modified version of XP called Simplified Extreme Programming (SXP) is proposed that eliminates some complexities of XP such as pair programming to simplify XP without affecting its agility. The various phases are explained without adequate details and the validation is also missing.

[13] The motivating reason is to have a new type of web frameworks which share the advantages of mockup driven engineering methodology with the ones of model driven engineering approach without forcing user to work on different tools.

There are no standard development models or practices for web application, thus organizations depend on the experience and knowledge of their team which is often a small group of developers. If a web application is developed using poor methodology, then the chances of its failure or low performance gets elevated. Many studies proffers that the major cause of web failures during operational and development phase is poor project management. Poor project management can even delineate the benefits of good engineering practices.

2.1. Earned Value Management

The project management technique used for evaluating the performance and progress of project in an objective manner is called earned value management [14]. Earned Value Management is also called earned value performance management and value project management. It cannot be denied that project planning and estimation is a complex and unavoidable process. The terminology used in VVM is mentioned below [15]:

$$\text{team velocity}(\text{hours}) = \\ = \sum(\text{finished story points in iteration} \times \\ \times \text{average number of hours per story point in project}) \quad (1)$$

Work capacity: As the sum of working hours spent in each iteration or Sprint. Work capacity represents the average cost per iteration.

$$\text{work capacity} = \sum \text{working hours in iteration} \quad (2)$$

Focus factor. It is defined as the team velocity measured in hours divided by the work capacity. Focus factor is used to predict if the team is running under or over its forecasted capabilities. It also represents relationship between velocity and the dedicated working hours.

$$\text{focus factor} = \text{team velocity}(\text{hours}) / \text{work capacity} \quad (3)$$

From the reviewed literature it can be said that the process of web application development is a complex and multidisciplinary approach. Technical Debt Management is a neglected issue and a dedicated research is required to identify and test the practices which can prevent technical debt. The existing agile development models are not capable of assisting teams in gathering correct or Actual requirement and scope of web application from the client.

3. PROPOSED MODEL

The proposed model is named 'Agile with Requirement and Quality Inspection (ARQI)' as the emphasis during the development cycles on maintaining quality in each phase and correct interpretation of client requirements. The proposed model for web application development is different from traditional approaches in many aspects. The distinctive features of the proposed model are:

- Inclusion of Requirement and Quality Inspector (RQI).
- Design according to standards.
- Quality gateway for coding.
- Checking for scope creep and domino effect in subsequent iterations.

One factor of novelty is the inclusion of Requirement and Quality Inspector (RQI). During the literature review and analysis of conducted survey it was found that, a large number of iterations result because of unclear requirement gathering. The client generally has a vague idea about the requirements. Expectation mismatch occurs even after drafting of SRS. The effect of the unclear requirement objectives gets magnified during design phase.

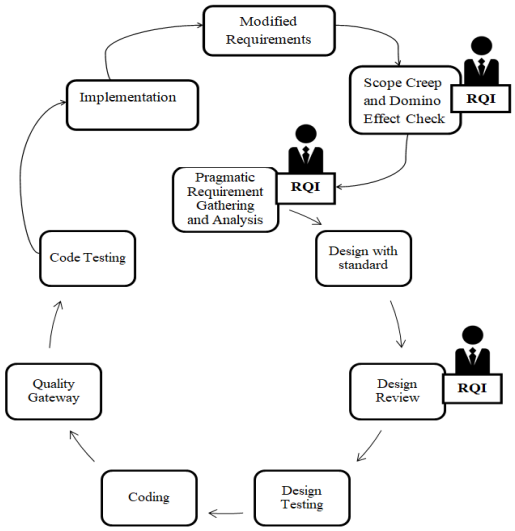

Figure 1. ARQI Model [16]

3.1. Pragmatic Requirement Gathering

Identifying user's needs is the foremost step in the project induction. In a traditional agile methodology client sends his initial requirements to the project manager related to the project. The development team is required to break down the requirements in to the scope of work.

3.2. Requirement Analysis

Once the requirements are gathered the cost-benefit analysis is performed in the requirement analysis phase the objectives are analyzed for feasibility. The objective of this phase is to ensure that the requirements are clear, complete and unambiguous. In this phase the shared understanding between the different stakeholders is developed and it becomes clear 'What is going to be developed'?

3.3. Design with Standards

In any web application design is a crucial factor that determines its success. In this phase both appearance and structure design take place. The structure design usually represents the network of hyperlinks that makes navigation website feasible and simple. Whereas in appearance design the look and feel of the website are finalized keeping in mind the end-users and requirements of the client. To reduce the number of iterations we have specified some standard for design as mentioned below. These standards not only reduce the number of iterations but can also help designers to avoid common design mistakes.

3.4. Design Review

Once a design is finalized it is reviewed by requirement and quality inspector as he is indirect contact client and can convey the Expectations of the client clearly. If requirement and quality Inspector find expectation mismatch, he can again send the design to the previous face for improvement.

3.5. Design Testing

Once the design is approved by RQI it is passed to design testing phase. Testing design early is very important to avoid blunders in the last stage. In this phase both the structure and appearance of the designer tested for functionality and if some discrepancy is a found it the design is again sent back to the design phase.

3.6. Coding

In this phase, the backend coding is performed. For this firstly requirements are gathered. The requirements may be databases, train staff, technical specifications and software. After gathering the requirements designer convert the design into machine code. The output of the stage is the finalized design document.

Once the coding has been performed it passes through a quality gateway in the proposed model. Traditionally the subsequent phase after coding is testing. However, in the proposed model an additional phase is allocated 'To test code for quality'. The objective of this phase is to restrict bad coding practices to ensure a code that is both easy to read and testable.

Coding with quality is very important while developing a web application. A badly designed code such as one with a lot of memory leak sand customization increases the time in further iterations and changes and may even lead to technical debt. Bad coding practices lead to accumulation of bugs, memory leaks and internal errors over the various iterations and makes testing and maintenance a strenuous task.

To avoid this in the code is evaluated for quality in the quality gateway. The codes can bechecked for quality either manually or automatically by using static analyzers such as Helix QAC and Kloc work to monitor the code quality metric.

3.7. Code Testing

The developed code with essential quality characteristics is tested to check whether each and every line of code has been executed. The aim of this stage is to detect the bugs present in the source code and remove them. The most popular methodology for code testing is unit testing. In unit testing individual components of the code are tested. Unit testing can be carried out manually or by using automation tools. In case of discrepancies the code is sent to previous stage for corrections.

3.8. Testing

Before implementation it is necessary to test the web application for functionality, usability, compatibility and security. This is done to avoid wastage of time and effort and identify points of failure at an early stage.

3.9. Implementation

In this phase the developed web application is implemented firstly on local servers review for changes. If some changes are necessary or if the client wants to add some additional features the first phase of requirement gathering is revisited. This face also comprises of maintenance. Continuous monitoring of website is done in maintenance for ensuring that only the updated and most recent information and content is displayed. The possibility of incorporating future changes is also ensured.

Table 1. Code quality metric [16]

Parameter	Definition	Significance
Extensibility	The degree to which the code can be extended to incorporate future growth is called Extensibility.	Extensibility means that new features can be added and some obsolete features can be removed without disturbing the entire source code.
Maintainability	Maintainability denotes the ease of making changes in the code and the associated risks with these changes.	Maintainability of a code can be evaluated based on the time required for making changes or number of lines of coded added for the same.
Readability and Code Formatting	A code is said to be readable if it is easy to understand.	Proper formatting and indentations should be used to make a code readable. Developers must also include remarks or comments according to the standards of the programming language being used.
Clarity	A code is said to be clear and unambiguous if it is easy to determine what the code is doing. A piece of code which requires a long time for interpretation then it said to be unclear and ambiguous.	Code must be well documented and follow a structured format so that it can be easily modified by other developer without going into the details of underlying functions.
Efficiency	A code that consumes only the required resources and executes in the least possible time is called efficient.	The following problems can arise if the code is not efficient: Longer build time. Complexity in bug detection and removal. Performance overhead.

3.10. Identifying Scope Creep or Domino Effect

If subsequent iterations client wants to add additional features or list some changes, then the RQI first check scoop creep and Domino effect. Organizations readily agree for adding an extra feature for consumer delight and future relations. However, it should be noted that adding extra feature such as a new filter in e-commerce increases the workload tremendously. Also, some changes may completely change the scope of the web application. To prevent such situations which lead to technical debt, an analysis for scope creep and Domino effect is compulsory. If the added functionality leads to either score crab or Domino effect the RQI redraft the budget and communicate today client.

In the final iteration the application is deployed on Wide Area Network.

Implementation: This is the last stage, in which the target users are made aware about the presence of the web application. Different promotional and marketing strategies are used for this purpose. In this step the web application is also monitored and issues are rectified.

4. IMPLEMENTATION OF PROPOSED MODEL

The proposed iterative lifecycle model by the name 'Agile with Requirement and Quality Inspection (ARQI)' is implemented during development of a web application based on Internet of Things (IOT). The application was developed within Tishitu, a part of MNIT Innovation and Incubation Centre, Jaipur, India. The authorities agreed to provide an insight of the development process being used in their organization through a pilot experiment. Additionally, the authorities also agreed to test the proposed model by implementing it on an additional pilot experiment.

Thus, the pilot experiment was conducted for developing web application related to IOT the development methodology followed by ARQI. One experiment was conducted using the ARQI approach. The details of experiment is mentioned in case study . It is important to note that all the economic data presented in this report are pure estimations for management purposes and does not represent actual economics of the company.

The case study the teams are formed on similar basis. The teams consist of likely skilled persons with similar experience level. The teams have access to equal resources. The ARQI model was implemented in the website 'www.tishitu.net' during the development of an internet of things application.

The observations at the end of each case study are recorded for qualitative analysis.

4.1. Case Study for ARQI

Using ARQI a web application case study is developed. Case Study's details is mentioned as per the minutes of meeting between RQI, team leader, team and/or client. The time period for each iteration was short and usually each iteration lasted for twelve days. The team comprises of:

- Team leader
- RQI
- Designer
- Coder
- Tester
- Phases are disused
- **Requirement Gathering:** The essential questions with keywords 'Why', 'What' and 'How' were answered with consent of RQI, team leader and client.
- **Requirement Analysis:** In this Phase Function and non-function requirements are summarized.
- **Design:** The appearance design was performed as per the standards. The design platform of word press is used for developing the web application. After the completion of appearance and structure design it was reviewed firstly by RQI then passed to testing. Four iterations were involved in this phase.
- **Design review:** The finalized design by graphic designer was passed to RQI for review.
- **Design Testing:** Once the design is approved by RQI it is passed to design testing phase. In this phase both the structure and appearance of the designer tested for functionality and if some discrepancy is a found it the design is again sent back to the design phase.
- **Coding:** Once the design was finalized the necessary backend coding was performed. The coding was checked for quality then passed for testing. This made changes easier and agile in subsequent iterations. MySQL database was used in the web application. The code was developed in CSS and html.
- **Code Quality Testing:** Once the coding has been performed it passes through a quality gateway where Code is checked for quality. The structure of code, no memory leaks and remarks were ensured. This stage facilitated easy understanding of code by the programmers who are not part of the current project team, for easier team shifting, if necessary.

• **Testing:** Firstly, the code is tested by usability testing. If no errors exist in source code then the web application is tested for the following tests: Functionality testing, Usability testing, Interface testing, Database testing, Performance testing, compatibility testing, Security testing

• **Implementation:** The developed web application was firstly implemented on local server "Xamp". Minor changes suggested by client were incorporated after analysis by RQI.

The number of story points to be completed is 56. The predicted number of iterations is 4. The Each iteration should not last for more than 200 hours. The planned working hours in each iteration are 119. All data show in Table 2.

Table 2. Initial forecast or iteration 0 estimates case study for ARQI

Initial Forecast	
Number of story points	56
Total Available Hours	654
Hours Per story point	10.9
Planned working hours per iteration	119
Predicted Number of iterations	4
Initial Team Velocity (Predicted in story points)	15
Cost per iteration	50,000 INR
Uncertainty	±16 %

5. DATA ANALYSIS

The major figures of project are shown in Table 3. Based on these figures the Earned Value Metrics is formed as shown in Table 4. The parameters used for EVM are Team Velocity (TV), Work Capacity (WC), Focus Factor (FF), Target Value Increase (TVI+), Expected Percent Completed (EPC), Actual Percent Completed (EPC), Planned Value (PV) and Earned Value (EV).The details of these parameters are mentioned in literature review. The project started with 56 story points, 3 story points were added in third iteration and one-story point was removed in fourth iteration. The project burns up chart is shown in Figure 1. It can be seen that the development proceeded with less fluctuations in number of story points. Other important aspects of the development with ARQI are explained with Figure 1.

Table 3. Major figures of project

Iteration	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Date of Start	03/06/19	15/06/19	27/06/19	7/06/19	17/06/19
Date of End	14/06/19	26/06/19	6/07/19	16/07/19	28/07/19
Working Days	12	12	12	12	12
Estimated Velocity	15	11	16	16	14
Real Velocity	7	10	10	17	14
Finished Story Points	7	17	27	44	58
backlog Story Points	56	56	59	58	58
Remaining Story Points	49	32	32	14	0
Estimated Working Hours	119	119	119	119	119
Real Working Hours	60	70	60	59	60

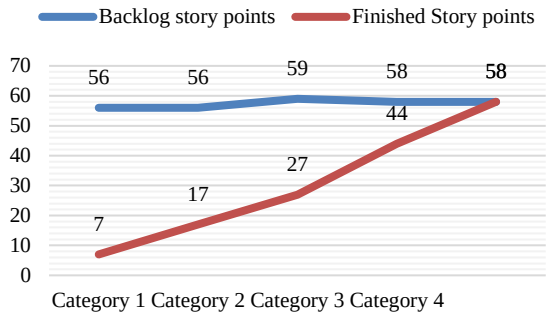

Figure 2. Project burn up chart

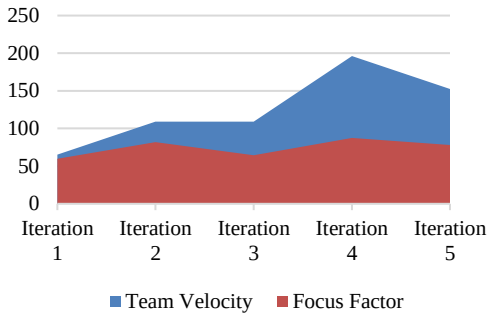

Figure 3. Trend of team velocity and focus factor in project

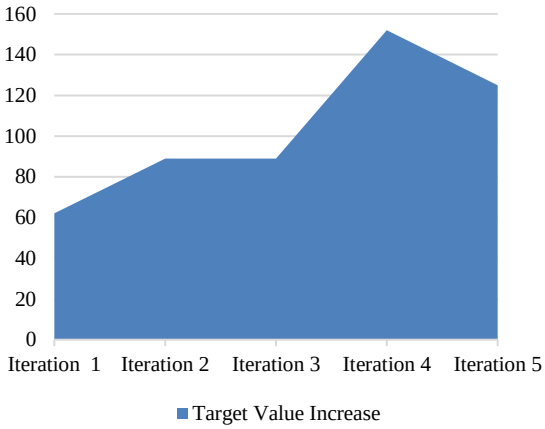

Figure 4. Trend of target value increase in project

From Figure 3 it can be observed that the project started with less team velocity which increased in subsequent iterations. The focus factor also has a positive growth trend and mostly remained greater than 100, thus denoting an efficient team. The TVI+ also showed a positive growth trend, as can be seen from Figure 4.

Table 4. EVM metrics of project

Iteration	AH/SP	Work Capacity	Team Velocity	Focus Factor	TVI+	EPC	APC	PV	EV
1	10.9	119	65.4	60	0.62	0.27	0.125	67,500	31, 250
2	10.9	119	109	81.9	0.89	0.32	0.30	80,000	75,000
3	10.9	119	109	64.68	0.89	0.55	0.46	1,37,500	1,15,000
4	10.9	119	196.2	87.5	1.52	0.74	0.76	1,85,500	1,90,000
5	10.9	119	152.6	78	1.25	1.00	1.00	2,50,000	2,50,000

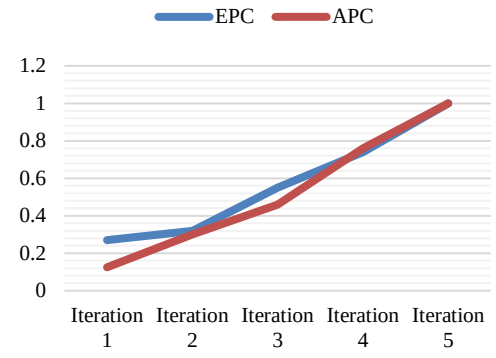

Figure 5. Estimated Percent Complete (EPC) vs Actual Percent Completed (APC)

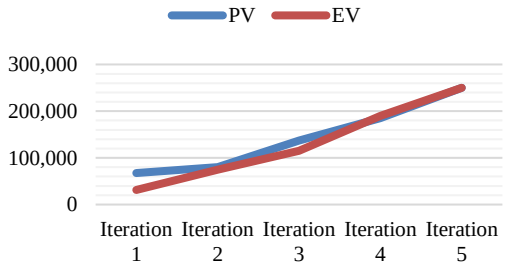

Figure 6. Planned Value vs Earned Value (EV)

The relation between EPC and APC can be seen in Figure 5. It can be seen that APC closely follows the trend of EPC. A similar relation is observed between PV and EV in Figure 6. This signifies that the deviations in project from planned or estimated parameters are less.

6. CONCLUSION

The project was completed with increased client and team satisfaction. We found a minor percentage of underestimation (20 percent) and coordination between the different stages of development improved because of the presence of RQI, who acted as a collaborating channel. Using this model, the number of iteration and the time in iterations was reduced. As the design was according to standards, changes were possible even with switching of team members. We found some minor managerial issues arise due to inclusion of RQI.

REFERENCES

[1] J. Williams, "Avoiding the CNN Moment", IT Professional, Vol. 3, No. 2, pp. 72-70, April 2001.

[2] D. Verdon, "Security Policies and Software Developer", IEEE Security & Privacy, Vol. 4, No. 4, pp. 42-49, August 2006.

[3] K.P. Birman, "CORBA: The Common Object Request Broker Architecture", Reliable Distributed Systems: Technologies, Web Services, and Applications, Vol. 1, No. 1, pp. 119-140, 2005.

[4] S.R. Ashumova, M.H. Zeynalova, R.B. Rustamov, "Space Technology Applications in Environment of Integration Engineering and Management", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 25, Vol. 7, No. 4, pp. 62-65, December 2015.

[5] T.H. Dostalizade, R.N. Mehdizadeh, A.V. Bannikova, E.T. Abdullayev, "Engineering Design Problems", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 18, Vol. 6, No. 1, pp. 183-186, March 2014.

[6] H. Gellersen, R. Wicke, M. Gaedke, "Web Composition: An Object-Oriented Support System for the Web Engineering Lifecycle", Computer Networks and ISDN Systems, Vol. 29, No. 8-13, pp 1429-1437, 1997.

[7] K. Chang, P. Henderson, "A Practice of UML for Web Development", R. Hamid (Ed.) Proceedings of the International Conference on Software Engineering Research and Practice (SERP), Las Vegas: Semantic Scholar, p. 1, 2012.

[8] F. Maurer, D. Wells, "Extreme Programming and Agile Methods - XP/Agile Universe", Verlag Berlin: Springer, 2003.

[9] T. Margaria, B. Steffen, "Agile IT: Thinking in User-Centric Models", T. Margaria, B. Steffen (Eds.) Leveraging Applications of Formal Methods, Verification and Validation, Berlin: Springer, pp. 490-502, 2008.

[10] E. Falzone, C. Bernaschina, "Model Based Rapid Prototyping and Evolution of Web Application", T. Mikkonen, R. Klamma, J. Hernandez (Eds.), International Conference on Web Engineering. Cham: Springer, pp. 496-500, 2018.

[11] B.L. Romano, A.M. Cunha, "An Agile and Collaborative Model-Driven Development Framework for Web Applications", S. Latifi (Ed.), Information Technology-New Generations. Cham: Springer, pp. 383-394, July 2018.

[12] D. Lloyd, R. Moawad, M. Kadry, "A Supporting Tool for Requirements Change Management in Distributed Agile Development", Future Computing and Informatics Journal, Vol. 2, No. 1, pp. 1-9, June 2017.

[13] M. Angelaccio, "Meta Page - A Data Intensive MockupDD for Agile Web Engineering", M. Helfert (Ed.) 12th International Conference on Web Information Systems and Technologies, Italy: Webist, pp. 315-317, 2016.

[14] H. Erdogmus, "Tracking Progress through Earned Value", IEEE software, Vol. 27, No. 5, pp. 2-7, 2010.

[15] C.J. Torrecilla Salinas, J. Sedeno, M.J. Escalona, M. Mejias, "Estimating, Planning and Managing Agile Web Development Projects under a Value-Based Perspective", Information and Software Technology, Vol. 61, No. 1, pp. 124-144, 2015.

[16] A. Kalwar, R. Ajmera, C.S. Lambaz, "Development of New Iterative Life Cycle Model", Thesis, JECRC University, Jaipur, India.

BIOGRAPHIES

Anju Kalwar was born in Kota, Rajasthan, India. She received the B.C.A and the M.Sc. degrees from University of Rajasthan Vidhayapeth, India and the Ph.D. degree from JECRC University, Jaipur, India. She has 7 research papers in prestigious journals and peer-reviewed conference proceedings.

Reema Ajmera received PGDCA, M. Tech., M.Sc. (CS) and Ph.D. (CS). Currently, she is an Associate Professor at JECRC University, Jaipur, India. Her research area is in software engineering. She has 8 research paper in prestigious journals, 8 research paper in peer-reviewed conference proceedings and authored 1 book.

Amandeep Gill received his B.E. from the University of Rajasthan, Jaipur, India, M.E. from Thapar University of Patiala, Punjab, India and Ph.D. from JECRC University, Jaipur, Rajasthan, India. He is working as Assistant Professor at JECRC University. He has eight and half years of teaching experience. His research fields are Distributed generated systems, power systems, artificial intelligence, and machine learning. He has authored 4 books and has 14 research publications in SCI Journals, Scopus journals and peer-reviewed conference proceedings.