How to Cite:
Deepa, T., & Thaddeus, S. (2022). Enhancing agile process with open unified process framework (OUPF). *International Journal of Health Sciences*, *6*(S2), 2034–2044. https://doi.org/10.53730/ijhs.v6nS2.5429

Enhancing agile process with open unified process framework (OUPF)

T. Deepa
Research Scholar, Thiruvalluvar University, Serkadu
Email: margaretdeepa23@gmail.com

S. Thaddeus
Assistant Professor, Department of Computer Science, Don Bosco College, Yelagiri Hills, Tirupattur
Email: thad@boscoits.com

***Abstract*---**The awareness and practice of the agile process has increased very much in recent years. Yet, there are gaps among the developers in agile adoption. A backup support from conventional process standards can enhance the outcome of agile work. Choosing Open Unified Process framework (OUPF) as process repository, a semantic model is proposed to link an agile software project with standard process components and guidelines. Any agile software project can be instantiated in terms of requirements (as user stories) and tasks using this semantic model, drawing guidance from OUPF. This semantic model is applied on a real time project in an industrial environment. The results show that there is an improvement in project outcome when agile is combined with Open Unified Process Framework.

***Keywords*---**agile process, open unified, process framework, requirement engineering, ontology, semantic web.

Introduction

There are different types of software process models in vogue. Agile model is the popular and active model that the industry practitioners adopt today. Literature of agile upholds it as healthier than conventional process models. However, there are some demerits in agile which can be resolved by including standard process components. This requires a framework which fits the conventional process elements into an agile process without compromising on agility. In this paper, a standard process model such as Open Unified Process Framework (OUPF) as an ontology is connected to the agile model for process enactment using semantic web standards and tools. Section 2 of this paper compares agile and conventional

process models taking requirement engineering as a sample. Based on the methodology explained in Section3, the semantic model and related ontologies are described in section 4. An experiment is shown with a real-time project to compare the effectiveness of project management with agile alone and agile with support from OUPF. Section6 concludes that mapping the tasks of agile with OUPF-backed up agile enhances the efficacy of the agile development model.

Literature Study

Literature on agile processes has comparisons between conventional and agile models from various perspectives [1,2,3]. *Batool et al.[1]* compared the key process elements namely role, activities and artefacts in the context of requirement elicitation and analysis. There is a strong indication that agile requirement engineering is healthier than the traditional process. Agile process is superior when mapped on factors such as project duration, risk analysis, flexibility, testability and customer interaction. Additional requirement documentation is proposed as a solution to the constraints of agile requirement engineering [2].

*Inayat et al.[3]*emphasized the importance of process, tools and documentation in applying agile models. The challenges faced in using agile requirement engineering are listed as lack of documentation, lack of methodologies to handle non-functional requirements and inappropriate estimation. The requirement specification is documented as test cases in [4] to improve the requirement traceability. While *Bjarnason et al.* found advantages in this methodology like better communication, effective change management and customer support, there is a gap in elicitation, verification of quality requirements and change management. *Goyal and Ramesh [5,6]* identified incomplete non-functional requirements and poor change management as areas of concern in agile requirement engineering. The conclusion of Goyal's study was that the analysts require sound domain knowledge and the skills to impart the knowledge to the development team.

The review of literature converges on the following areas for improvement such as Comprehensive Documentation - Requirements captured just in time for progress and documented with less details. This is inadequate for the fresh entrants into the project. It can lead to a misunderstanding and difficulties to define and take up new features [3, 9]. There is a need for comprehensive documentation.

Non-functional requirements - Requirement validation is done by face to face communication [8]. The focus is more on whether the requirements satisfy the user's needs rather than on its correctness and completeness. The consistency and completeness of the requirements are not formally checked as in conventional documentation [4, 9]. The least focus is on non-functional requirements. When the documentations are not clear and the tasks are not specific and comprehensive, it leads to a mismatch in cost or effort estimation. The developers need to organize requirements throughout the project life cycle for imprecise understanding and mismanagement of the project.

Proposed Technique

The objective of this proposed work is not merely to analyze the superiority of the agile model compared with the conventional but enhancing the productivity in the agile model with the strengths of conventional models. The agile model has limitations with regard to documentation and non-functional requirements. These conditions increase due to avoidance of many tasks during process implementation. The scenario is more critical when the development team is heterogeneous in terms of experience and competency. Agile processes are generic while standard process models like OUPF are prescriptive. The roles and tasks to be done are specific and it provides direct enactment by the developers in OUPF systems. It leads to better cost and effort estimation. Such a prescriptive process model can be anchored to an agile project repository. OUPF is chosen as the prescriptive process model. It can be linked to any agile project as a process ontology.

OUPF provides the components for process implementation such as: endeavors, stages, producers, work units, work products, languages and usage guidelines [6]. OUPF is designed as three components namely meta model, components repository and guidelines for construction and usage. The Meta model defines the fundamental reusable method components namely process elements. Component repository gives the actual description of all reusable methods while Construction and Usage Guidelines state how to reuse the method components in situation-specific. The OUPF repository is exhaustive and addresses all scenarios of process implementation. Based on the situation, it could be tailored. This is suitable to give prescriptive tasks for the developers when generic tasks are listed down in an agile sprint based on product back-log.

A semantic model is suggested to link OUPF with any agile process. OUPF is linked to any agile project through an agile process framework. Every task listed in a project sprint is elaborated with tasks from OUPF. This will facilitate the developers to be comprehensive and specific in development activities. More precise estimation and requirement management can be obtained.

Semantic Model

Earlier research [7] on Semantic Enabled Software Engineering (SeSEE) developed a semantic model for software process management. This model is based on Description Logic combined with Semantic web standards. The knowledge base is defined as K= (T, A, R) where T denotes terminologies, A denotes assertions and R denotes rules. The rules are represented as C => D that if an element e In Figure 1, represents the ontology used to represent the terminologies and assertions of a domain where TBox covers Terminologies, Abox covers assertions and Rbox comprises the rules of domain. Three operations make the knowledgebase usable. TELL is to build or modify the knowledgebase. ASK is for retrieval of information. ACT is used as an additional operation to extend knowledge base for automation.

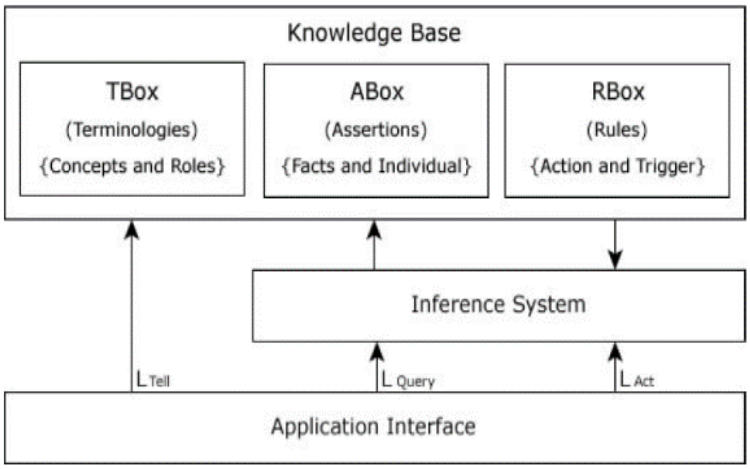

Figure 1 – DL Based Knowledge System with Extended Operations.

This semantic model is implemented using OWL-DL, semantic web language, promoted by World Wide Web Consortium. OWL-DL reduces the inherent constraints in representing knowledge using DL and makes it appealing for the end users. OWL-DL represents the process knowledge with regard to OUPF or agile model as terms, assertions and rules. Thus, two distinct ontologies are developed for OUPF and agile process framework. Ontologies are developed using Protégé, an open source ontology editor [7]. These ontologies are mapped by the common terminology: Task. Every agile project is instantiated using the agile process framework ontology. Task in sprint Backlog of agile project is linked with task of OUPF repository where ever required and feasible to provide process-specific guidance to the developers. Inference system of the OWL-DL engine provides the most fitting support for the developers. The application interface is linked to any project management tool of a software development environment.

Ontology For OUPF

OUPF ontology represents the metamodel elements namely Guidelines, WorkPlan, Role, Activities, Process, Tasks and Steps. Using this conceptual schema, details of process enactment are stored as assertions.

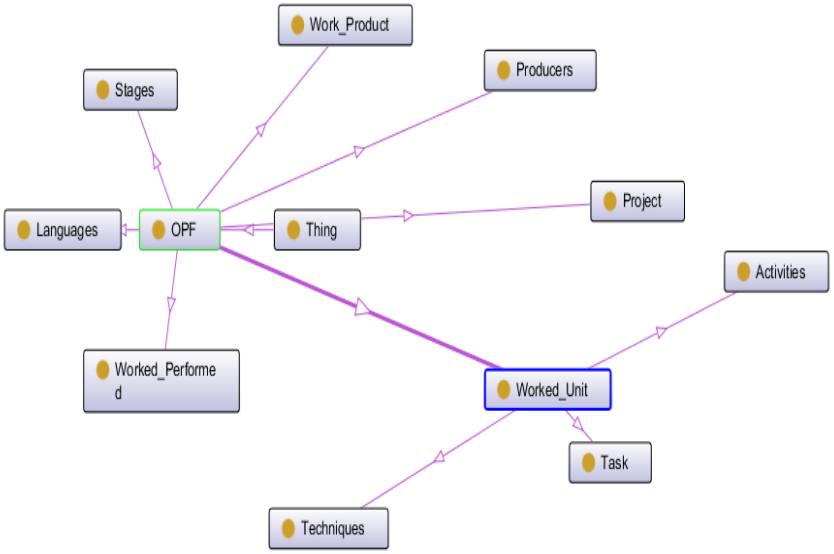

Figure 2 – Open Unified Process Framework Ontology

In Figure 2, all the meta-model elements are method components. The component which produces anything that is valid is called Work Products. These are the products produced by producers while some performance of Work units. Activity, Task and Techniques are the three kinds of Work Units. The section which is used to document the work product is called language. Endeavors are the module which is further divided into projects. The duration refers to time taken by the work unit to produce the product delivery. Work Performance is the section which models the work units by producers.

Agile Process Ontology

The key process elements of an agile model are : project, user stories, sprint, deliverables, and tasks. Every project is stored as an instance or assertion with its distinct user stories. These user stories are connected to the sprints and deliverables and expressed in terms of tasks to be done.

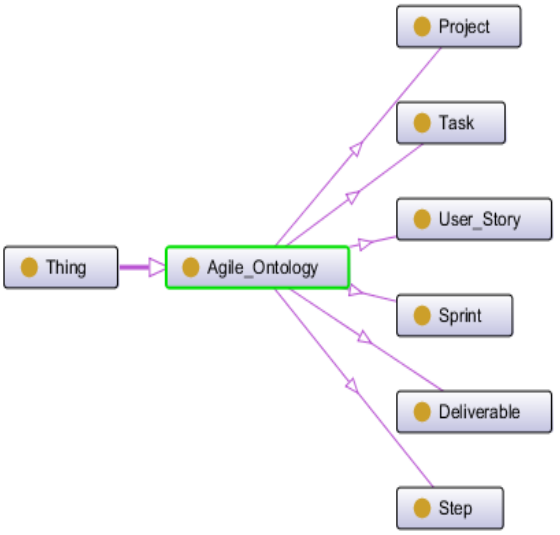

Figure 3 – Ontology for Agile Software Development

A project has many sprints and every sprint is assigned with a list of user stories. Sprint is also associated with a set of tasks and deliverables.

Mapping Agile And OUPF

Figure 4, explores the mapping of Agile and OUPF. This mapping is achieved by mapping TASK in an agile model to TASK in OUPF with an object property as OWL-DL allows. Mapping possibilities are presented to the end user using the inference mechanism of ontology-driven knowledge systems. The tool developed for software process management using semantic web in the research work is used for this mapping purpose [7]. It enables the developers to understand Activities, Work Plan, Guidelines and Role in a better way. It helps them to estimate effort better and thus gain improvement on project management.

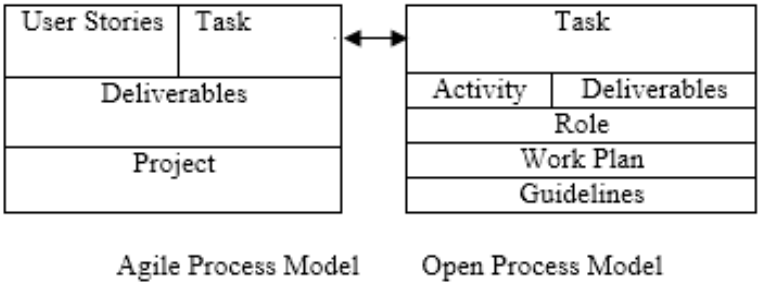

Figure 4- Mapping Agile and OUPF

Experimental Validation

SmartSchool+ is a software product, developed and supported by a dedicated software development team in BoscoSoft Technologies Pvt Ltd. This project is chosen for the experiment. The developers adopt an agile process. Earlier two sprints are taken and their burn-down chart is studied with their product backlog. Burn-down chart is the tool which helps to calculate the development of the project by plotting the number of days of sprints against the number of remaining working hours to complete the project. This helps the developers to verify whether they are in progression with the software development.

Phase – I: Sprint Backlog(1) - Using Agile model Development

There are 5 user stories in this sprint. The developers have taken 14 days to complete the sprint. Here the ideal time is 380 hours to complete the project. The Ideal hour represents the estimated hours to complete the sprint by the developers.

U ID	Task	Status	Initial Estimate of effort	1	2	3	4	5	6	7
1	Update User stories and Task for 1.47.0	Completed	48	24	24	0	0	0	0	0
1	Study the impact of Multiple Acyear in SSP	Completed	16	16	16	16	16	16	16	16
1	Distribution Register - (Logic Change)	Completed	5	5	5	5	5	5	5	5
2	Existing Stock Register - (Logic Change)	Completed	5	5	5	5	5	5	5	5
3	Individual Distribution - (Logic Change)	Completed	5	7	7	7	7	7	7	7
4	Design Inventory Receipt for SFSBLR	Completed	1	1	1	1	1	1	1	1
5	Implement Method to load the individual receipt	Completed	4	4	4	4	4	4	4	4
6	Implement Fee Head Mapping	Completed	3	3	3	3	3	3	3	3
8	Unit test and Verification	Completed	1	1	1	1	1	1	1	1
9	Testing	Completed	5	5	5	5	5	5	5	5
1	Report - Stock In Need Design	Completed	1	1	1	1	1	1	1	0
2	Implement methods to bind the report	Completed	3	3	3	3	3	3	3	0
8	Testing	Completed	3	3	3	3	3	3	3	3
1	Design page to get the credits from the user(Admin of SSP)	Completed	1	1	1	1	1	1	1	1
2	Write methods to fetch the details by particular school code	Completed	2	2	2	2	2	2	2	2
3	Generate the encrypted string for School code,SMS Credit and Random string	Completed	2	2	2	2	2	2	2	2
4	Upload file through web application, and match the settings with online and update the details	Completed	2	2	2	2	2	2	2	2
5	Unit test and Verification	Completed	1	1	1	1	1	1	1	1
6	Testing	Completed	2	2	2	2	2	2	2	2
			380	355	329	304	279	253	228	203
			380	376	369	334	322	302	288	250

Table. 5.1 Data table for plotting Burndown Chart

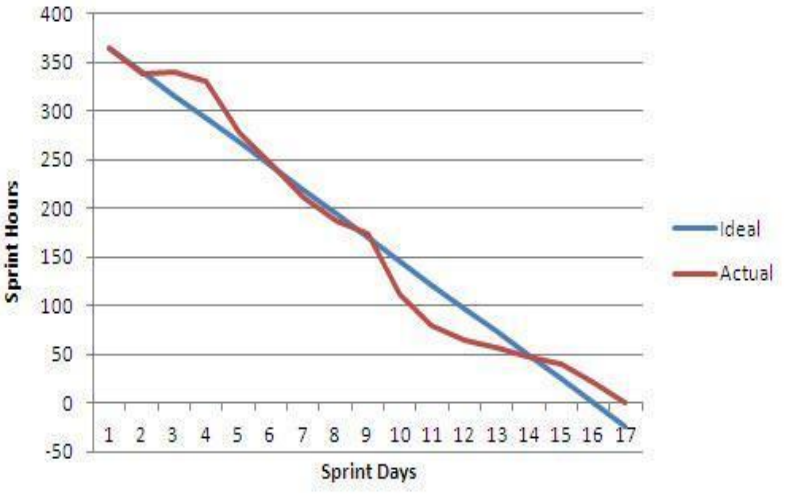

Figure. 5.2 Burn Down Chart for the Data of Table 5.1

The figure 5.2 clearly shows that from the Day1 to Day2, The Ideal and the actual plots are fallen in the same line which shows that the developers are on track in working. From the Day2 to Day6, The Actual line is above the Ideal line which shows that the developers are behind the schedule. During the Day9 to Day14, the Actual line is below the Ideal line and hence it shows that the team is ahead the schedule during those days. But the chart exhibits that the developers are behind the schedule from the Day14 to Day17, in their development work

Phase – II Sprint Backlog (2) - Using Agile Model Development

There are 6 user stories in this sprint. The total estimated work hour for this sprint is 295 hours.

UID	Task	Status	Initial Estimate of effort	1	2	3	4	5	6	7	8	9	10	11	12	13
	Writing of User Story	Completed	16													
1	Design TC	Completed	5	5	5	0	0	0	0	0	0	0	0	0	0	0
2	Design Study Certificates	Completed	1	1	1	0	0	0	0	0	0	0	0	0	0	0
3	Design Whomsoever Certificates	Completed	1	1	1	0	0	0	0	0	0	0	0	0	0	0
4	Construct query to bind the certificated	Completed	4	4	1	3	0	0	0	0	0	0	0	0	0	0
5	Assign parameter values to certificates	Completed	1	1	1	1	0	0	0	0	0	0	0	0	0	0
6	Unit Test and Verification	Completed	2	2	2	2	0	0	0	0	0	0	0	0	0	0
6	Testing and Verification	Completed	5	5	5	5	0	0	0	0	0	0	0	0	0	0
1	Provision to enter marks for multiple subject (instead of Combo provide checkbox) in Examination Mark Entry page	Completed	16	16	16	16	16	16	16	8	8	0	0	0	0	0
1	Testing and Verification	Completed	8	9	9	9	9	9	9	9	9	9	9	0	0	0
1	Fix the Problem in Registering Onduty and Permission for same day and same session	Completed	8	8	8	8	8	8	8	0	0	0	0	0	0	0
1	Performance tuning of new portal	Completed	11	11	11	11	11	11	11	11	11	11	11	11	11	11
1	Testing and Verification	Completed	3	3	3	3	3	3	3	3	3	3	3	3	3	3
1	Provision to view student Annual progress card in Parent Login	Completed	16	13	13	13	13	13	13	13	13	13	13	13	13	31
2	Teasting and Verification	Completed	4	4	4	4	4	4	4	4	4	4	4	4	4	4
1	Implement Corpus Fund Concept	Completed	16	16	16	8	9	0	0	0	0	0	0	0	0	0
1	Testing and Verification	Completed	4	4	4	4	4	0	0	0	0	0	0	0	0	0
	Uninterrupted Working Hours - Ideal		295	275	256	236	216	197	177	157	138	118	98	79	59	39
	Uninterrupted Working Hours - Actual		295	277	274	261	249	216	203	168	161	133	##	89	77	87

Table 6.1 Data table for plotting Burndown Chart

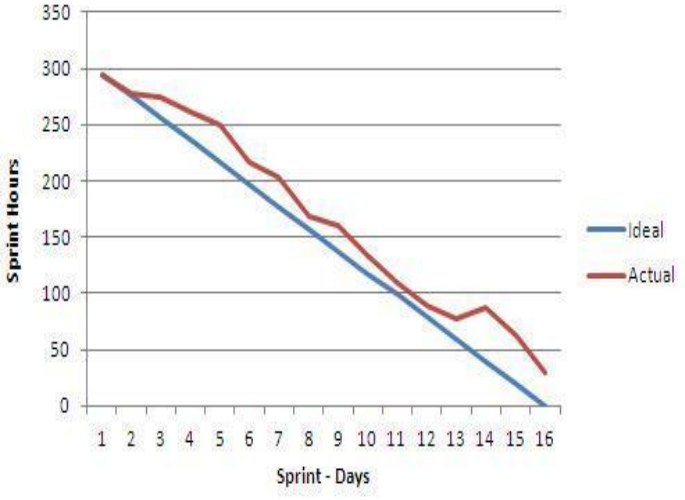

Figure. 6.2 Burn Down Chart for the Data of Table 6.1

Figure 5.2, clearly shows that from Day1 to Day2, the progress is as per schedule. But from the Day2 to Day16, the Actual line is above the Ideal line which means that the number of remaining hours of working is high and the days alone burned down which reveals that the progress of the development is behind the schedule. It feeds back the developers that they must increase the speed of their work or to act differently.

Phase – III Sprint Backlog(3) - Using OUPF Semantic Model

The developers are trained to use the semantic web tool which provides process guidance on every task. 7 user stories were in the sprint. The developers took 20 days to complete the sprint. Here the estimated hours were 366 to complete the project.

U ID	Task	Status	Initial Estimate of effort	1	2	3	4	5	6	7
	Update User stories and Task for 1.45.6	Completed	26	26	26	0	0	0	0	0
1	Documentation of GPS configuration and Test Connection	Completed	1	1	1	1	1	0	0	0
1	Documentation on GPS Entrypt	Completed	4	4	4	4	4	4	4	4
2	Study and Design the Table Structure	Completed	1	1	1	1	1	1	1	1
3	Read data from GPS	Completed	20	4	4	4	0	3	8	0
5	Implement Format method for the Return Value	Completed	2	2	2	2	2	2	2	2
6	Store the result in DB from device return format to Langtitude and Longtitute	Completed	2	2	2	2	2	0	0	0
7	Code Review and Testing	Completed	1	1	1	1	1	0	0	0
8	Testing	Completed	12	12	12	3	3	3	3	0
1	Study on Display read data from GPS	Completed	20	5	5	5	5	5	0	0
2	Testing	Completed	4	1	1	1	1	1	1	1
1	Mannually check with different Service Provider to get the return string status	Completed	2	2	2	2	2	2	2	2
3	While Process Consider the "-1" as fail for the components	Completed	6	2	2	2	2	2	2	2
4	Change the All report query to show absent	Completed	8	1	1	1	1	1	1	1
5	Change the Exam progress card query to get Absent List	Completed	8	1	1	1	1	1	1	1
6	Code Review and Testing	Completed	1	1	1	1	1	1	1	1
7	Testing	Completed	5	1	1	1	1	1	1	1
1	Write the method in St Claret School dll to get the exam mark	Completed	8	8	8	8	8	8	8	8
	Uninterrupted Working Hours- Ideal		366	344.5	322.9	301.4	279.9	258.4	236.8	215.3
	Uninterrupted Working Kours- Actual		366	307.5	307.5	272.5	252	242	203	180

Table 7.1 - Data table for plotting Burndown chart

Figure.7.2 BurnDown Chart for the data of Table7.1

Figure 7.2, showed that, after adopting the prescriptive model that is mapping the agile tasks with OUPF using ontology that is using the semantic web tool, the development is consistent and the developers can be ahead of the estimated schedule of working hours.

Conclusion

A single project alone cannot assert the base of the proposal. Additional of varying nature and complexity should be experimented to validate and bring out the effectiveness of combining OUPF with agility. As a further, research will be

complemented to focus on other problems on agile development of intricacy and large software systems, as well as, broadening the research to comprise other projects of different levels of complexity.

References

1. Asmabatool ,yasirhafeezmotla and Bushra Hamid, “Comparative Study of Traditional Requirement Engineering and Agile Requirement Engineering”, 15th International Conference on Advanced Communication Technology (ICACT), January 2013, pp. 1006-1014, Pyeongchang.
2. Ville T. Heikkil, Casper Lassenius, Daniela Damian and Maria Paasivaara, “A Mapping Study on Requirements Engineering in Agile Software Development”, in Proc. SEAA, 2015, DOI.10.1109, p.70.
3. IRUM INAYAT, SITISALWASALIM AND MAYA DANEVA, “A SYSTEMATIC LITERATURE REVIEW ON AGILE REQUIREMENTS ENGINEERING PRACTICES AND CHALLENGES”, IN COMPUTERS IN HUMAN BEHAVIOR, 2014, DOI.10/1016, P. 46.
4. Elizabeth Bjarnason and Michael Unterkalmsteiner, “Multi-Case Study of Agile Requirements Engineering and the Use of Test Cases as Requirements”, in Information and Software Technology, 2016, DOI. 10.1016, p.008.
5. VishvadeepTripathi, Arvind Kumar Goyal, “Agile Requirement Engineer : Roles and Responsibilities”, in Proc. IJISET, Vol. 1 Issue 3, May 2014.
6. D. Zowghi, D.G. Firesmith and B. Henderson-Sellers, “Using the Open Process Unified Framework to Produce a Situation-Specific Requirements Engineering Method”, Basic Concepts and Taxonomy of Dependable and Secure Computing (2004).
7. S. Thaddeus, “Semantic Web-Enabled Approach to Software Process Management” ,Ph.D thesis, Bharathidasan University, Tiruchirapalli, India, Apr. 2011.
8. Balasubramaniam Ramesh, Lan Cao and Richard Baskerville, “Agile requirements engineering practices and challenges: an empirical study”, in Info Systems J, 2010,vol.20, p. 449–480.
9. FraukPaetsch, Dr. Armin Eberlein and Frank Maurer, “Requirements Engineering and Agile Software Development”, in Proc WETICE’03, 2003.
10. Monika Agarwal and Prof.RanaMajumdar, “Tracking Scrum projects Tools, Metrics and Myths about Agile”, in Proc. IJETAAE, ISSN 2250-2459, Vol. 2, Issue 3, March 2012