ADVANCES in NATURAL and APPLIED SCIENCES

ISSN: 1995-0772
EISSN: 1998-1090
2016 April 10(4): pages 355-362

Published BY **AENSI Publication**
http://www.aensiweb.com/ANAS
Open Access Journal

Student-Driven Learning Circle through Software Development Life Cycle (SDLC) Methodology

[1]Pradhiba Selvarani M and [2]Hari Sankar S

[1]Assistant Professor, Computer Science and Engineering, Kamaraj College of Engg. and Technology, Virudhunagar, India.
[2]Technical Analyst, QAD, Thirdware Solutions Ltd., Chennai, India.

Received 25 January 2016; Accepted 18 April 2016; Available 28 April 2016

Address For Correspondence:
Pradhiba Selvarani M, Assistant Professor, Computer Science and Engineering, Kamaraj College of Engg. and Technology, Virudhunagar, India.
E-mail: mpradhiba@gmail.com

ABSTRACT

In an era where technology has grown by leaps and bounds the art of teaching students has remained stagnant. There is a huge disconnect between what is being taught in colleges and what a graduate faces in current IT industry. Students who graduate and get a job in an IT company find it increasingly difficult to cope with the scenarios and seem unprepared. The most important thing about teaching in college is to ensure that the students once they graduate are finished products whose potential can be utilized by the IT industry. The aim of this paper is to propose a method of teaching to simulate an IT environment in the campus so that students will be better prepared once they graduate and enter an IT company. The idea proposed is to match the student's life cycle with the Software Development Life Cycle.

KEYWORDS: Pedagogy, Software Development Life Cycle (SDLC), Requirement Analysis, Design, Testing and Validation

INTRODUCTION

In a normal education scenario in India, the students are trained to get marks and crack the exams. Although their marks establish their credentials it affects them in the long run. Only a handful of them are able to solve problems in a real-life scenario applying the methodologies taught to them in college. Not surprisingly, these are the ones who climb higher in the ladder of IT industry using their logical as well as technical prowess. The key here is to reduce the gulf between what a student faces in College and what he or she faces in an IT company.

IT companies follow a set of processes when they take up a project. They divide a project into different phases and provide key milestone dates and work towards achieving their targets phase by phase. The same phase-wise management can be followed in teaching [7] as well. Requirement gathering, Estimation, Analysis and Design, Implementation, Testing and Maintenance [1] can be the mantra based on which a lecturer can teach in college instead of being just a two-mark questions.

A. Problem Statement:

A subject usually consists of five units divided into various chapters with headings and sub-headings with the first unit mainly an introduction of the subject.

The timeline in which a lecturer can teach all these five units is very stringent which in turn make the inefficient teachers to skip some chapters with an excuse that these may not be so important. Whereas skipping not so important chapters may be time saving, the students when they prepare for the exams have to go through

To Cite This Article: Pradhiba Selvarani M and Hari Sankar S., Student-Driven Learning Circle through Software Development Life Cycle (SDLC) Methodology. **Advances in Natural and Applied Sciences**. 10(4); Pages: 355-362

every topics. The students are at the moment at the mercy of their lecturers. An inefficient teacher makes the work of a student even tougher.

Existing System:

Initially a timeline is given to the lecturers to finish each unit within a specified timeline. In an ideal world, the lecturer will complete a week before the exams and utilize the remaining time for review or giving the students some homework or problems to solve in class which will ensure that students will be ready when exams come. Instead, lecturers complete the class at the last minute and only few of the students will be prepared once the exams come. And once the exams are completed and marks are announced-there is usually no post-mortem of the results to ensure where each student went wrong.

Proposed System:

As you saw above, in the existing system the lecturer is the ringmaster and the students involvement is only in learning what the lecturers have given them. The method that we are proposing here is to make the student learn by involving them in the teaching methodology. This will ensure by practicing what they learn the process will be etched in their minds better and they will be ready to face the real-life scenarios and implement the techniques they have learned in a far more efficient manner.

Each subject has five units and each unit can be taken up as a project. The students are the resources who will be working in this project. Each unit will then be divided into four sub-topics. Meanwhile, the class will be divided into four teams each being allocated a topic they will have to deal with.

The proposed method uses Software Development Life Cycle [3] (SDLC) (as in Fig. 1.) as a methodology to be used in the pedagogy that we have implemented.

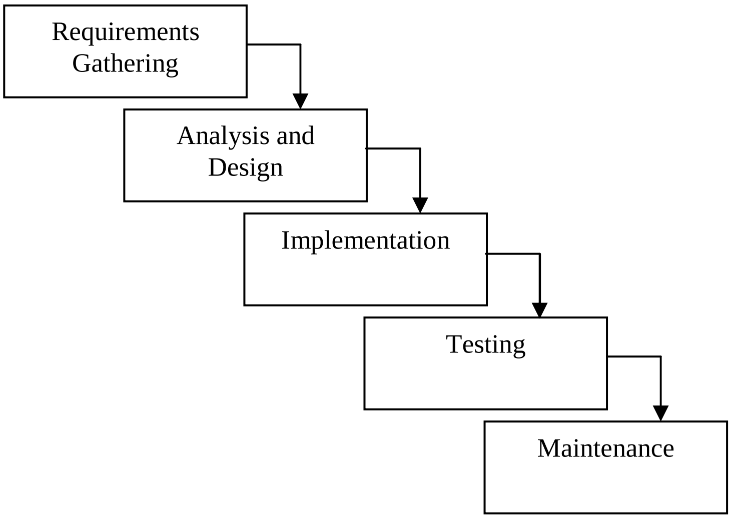

Fig. 1: Software Development Life Cycle

A. Project Definition:

The lecturer here is the Project Lead. On the first day, the lecturer will be provide an overview of what the chapter is about and the details they expect the students to read through. In a way this is the Project Kick-off underlining the scope that will be covered during the Project implementation. The lecturer will detail what is expected of the students when proceed with their presentation.

1) Requirements Gathering and Analysis:

The requirement gathering [2] is the key phase. In this requirement gathering phase the lecturer will act as the customer. Each topic will be covered in a day. The lecturer will provide a walk-through of the topic and the stage is set for the project resources to clarify their doubts in this phase. For easier access later on this phase will be audio recorded. This will ensure that during teaching the students get involved at the initial stage itself thereby it will be easier for them to get their roles defined correctly.

The questions that they come up will eventually help a lot when they proceed to plan for the Designing and the Implementation phase. The lecturers will initially go through the given topic and as per the usual format they will be presenting to the students the details that have been covered in a given topic. This will not be just an overview of the topic-it is the duty of the lecturer to ensure that the key points are covered in their presentation.

The students' role here is to understand the details provided by the lecturers and then clear their doubts immediately. This is similar to Knowledge Transfer in IT industry and at the same time this can be equated to

Requirement Analysis [6] phase-the phase where the customer provides in depth details of their requirements. The following Table I depicts this phase.

Table I: System Design of Requirements Gathering and Analysis Phase

Inputs	Outputs	Deliverables	Actors
Lecturer's walkthrough on each topic	Students' Question and Answer	Documentation of Question and Answer and Audio files	• Lecturer:- Customer • Students:- Project Team

2) Design and Implementation:

The project team-once requirement gathering phase is complete begins their analysis and plan how to proceed with implementing. Students are grouped into 4 groups and they will be given a walkthrough on the topic. The students will have to look at the requirements laid out by their lecturers and then plan how to proceed with design process. Based on the inputs provided by the lecturer in the earlier phase the students have to decide the best way they can design the output so that their presentation will be eye - catching as well as most effective.

There are multiple ways in which the presentation can be done. It can be in the form of power-point which can then be used for presenting the various points that the students need to put forth so that the rest of their classmates can understand clearly. It can be in the form of a skit-which will essentially help in effectively impacting the audience. For example-On a topic such as Information security-a skit/drama can go a long way in providing the audience an insight about the origin, impact and prevention of information hacking. Or it can be in terms of videos [10]-In the social media era almost any topic in the world can be Googled and information obtained easily. The most effective medium is the visual medium and videos can easily appeal as well as help in understanding the concepts easily. Audio files can help in detailing trivial details which will be helpful for short answer (two marks) questions. This phase is explained in detail as in Table II.

Table II: System Design of Design and Implementation Phase

Inputs	Outputs	Deliverables	Actors	End Customers
Documentation from the previous phase (Requirements Gathering)	Presentation of the Students	Videos, Charts, Flowcharts, Audio files, PPT presentations	• Presenting Students: Project Team	• The rest of the students / Lecturer

3) Testing and Validation:

This is a critical phase in the segment (as shown in Table III) as this is the phase that is used to assess how good the product is. There are two methods in which assessment can be done in this teaching methodology First mode of testing deals with testing the presenters. The listeners can put forth various questions in this case and ideally it will test how good they have covered the topic they have been given. Again-this will show their preparedness as well as showcase the attentiveness of the listeners. The more there is meaningful interaction the better it is for everyone concerned.

The next assessment is for the listeners. The presenters can provide a questionnaire at the end of their session to check how well the listeners have paid attention to the presentation. After the presentation is done by each group it is the role of lecturer to assess the presentation of each group and provide feedback. After the feedback is provided the usual unit tests can be given to the whole group.

This will be the integration Testing phase. The whole group will be taking the test on the whole unit and it will be easier to assess the capacity of the whole group as well and it will also help in understanding the efficiency of each presentation.

Table III: System Design of Testing and Validation Phase

Testing Phase - I

Inputs	Outputs	Deliverables	Actors
Questions / doubts from listeners	Answers provided by the presenters	Documentation of Queries / Answers	• Listeners:- Testing team • Presenters:- Project team

Testing Phase - II

Inputs	Outputs	Deliverables	Actors

Questionnaire from presenters	Answers from the listeners	Questionnaire	• Presenters:- Testing team • Listeners:- Reverse KT providing Support Team

Testing Phase – III (Integration Testing Phase)

Inputs	Outputs	Deliverables	Actors
Test questions from Lecturer / Examiner	Answers from Whole Class	Exam Questions and Answers along with results	• Lecturer:- Robust testing team • Class:- The whole system

Phases Description with Real – Time Input:

A. Project Description:

The project description as illustrated in Table IV initiates the process of Communication by the lecturer with the students as described in Table IV.1. Then, at the end of each session the students will ask queries on that topic and get it clarified by the lecturer which will also be shared as a documentation as in Table IV.2.

Table IV: Project Description

TABLE IV.1 Student Lecturer Communication	*TABLE IV.2 Queries with Solutions*
UNIT 1 - CONTENTS I. S/W Engineering Paradigm II. Verification & Validation III. Life Cycle Models Waterfall Model Incremental Process model Evolutionary Process model Specialized Process model IV. System Engineering Computer Based System The System Engineering Hierarchy Business Process Engineering Product Engineering V. System Modeling	UNIT-I SOFTWARE PRODUCT AND PROCESS **1.** Define Software Engineering? The application of a systematic, disciplined, quantifiable approach to the development, operation & maintenance of software (i.e.) the application of Engineering to Software. **2.** State the characteristics of Software? • Software is engineered or developed; it is not manufactured in the classical sense. • Software doesn’t wear out. • Although the industry is moving toward component based assembly, most software continues to be custom built. **3.** List out the activities of Linear Sequential Model. • Software requirement analysis (Understand the nature of the problem, information domain, function etc.) • Design (Translates the requirement in to Software representation.) • Code generation (Translates design in to Machine-readable form.) • Testing (Test the Logical internals functional externals of the System.)

B. Planning:

The next phase is the Planning phase (as in Table V), where the Estimation of Course Structure for a subject is detailed in Table V.1, the Scheduling of each task for the respective topics in each unit is portrayed as in Table V.2 and the Tracking of all the tasks completion is checked in parallel as described in Table V.3.

Table V: Planning

TABLE V.1 Estimation

Topic	Ref. / Text book	Page No.	No. of hours planned	Teaching aids OHP / PPT / Video / NPTEL / Animation / Model/ Chart	∑ ho
UNIT I - SOFTWARE PRODUCT AND PROCESS (9 hrs)					
Introduction S/W Engineer ng Paradigm	T1	53-58	1	PPT	1
Verification ,Validation	R2	535-57	1	NPTEL	2
Life Cycle Models Waterfall Model Incremental Process model	T1	77-83	2	OHP	4
Evolutionary Process and Specialized Process model	T1	83-93	2	--	6
System Engineering Computer Based System	T1	155-160	1	PPT	7
Business Process Engineering overview ,Product Engineering overview	T1	161-170	1	--	8
Overview- System Modeling	T1	164-170	1	OHP	9

TABLE V.2 Scheduling

S.No.	Register No.	Name	Topics Assigned
GROUP 1			
1	920411104083	SINDHUJA.M	Life C Models: Waterfall Model, Incrementa Process M
2	920411104084	SINDHUJA.R	
3	920411104095	TAMILSELVI.D	
4	920411104100	VAISHNAVI.R	
5	920411104102	VIDYA.T	
6	920411104111	YAMUNA RANI.S	
7	920411104301	ANITHA.N	
8	920411104314	RAJA SRI.M	
9	920411104323	VIJAYA LAKSHMI.J	
GROUP 2			
1	920411104004	ABUBUCKER.M	Evolutiona Process M Specialize Process M
2	920411104009	ARAVINDRAJ.R	
3	920411104010	ARUN KUMAR.P	
4	920411104016	BALASUBRAMANIAN.M	
5	920411104058	NAVEEN PRABU.E	
6	920411104063	PRABHAHAR.P	
7	920411104076	SAKTHI VIVEKANANDAN.U	
GROUP 3			
1	920411104086	SIVA PRASAD.J	System Engineerin Computer Based Sy Business Process Engineerin Product Engineerin
2	920411104107	VIJAYAKUMAR.R	
3	920411104109	VINOTHKUMAR.K	
4	920411104304	DHEIVAKARAN.P	
5	920411104308	KARUPPASAMY.M	
6	920411104317	SATHISHKUMAR.M	
7	920411104325	WILSON.K	

TABLE V.3 Tracking

S.No.	Register No.	Name	Topics Assigned	Planned (Topic)	Actual (Topic & (DD
GROUP 1					
1.1	920411104083	SINDHUJA.M	Life Cycle Models: Waterfall Model, Incremental Process Model	Waterfall Model	Waterfall Model (18/(
1.2	920411104084	SINDHUJA.R			
1.3	920411104095	TAMILSELVI.D			
1.4	920411104100	VAISHNAVI.R			
1.5	920411104102	VIDYA.T		Incremental Process Model	Incremental Process M
1.6	920411104111	YAMUNA RANI.S			
1.7	920411104301	ANITHA.N			
1.8	920411104314	RAJA SRI.M			
1.9	920411104323	VIJAYA LAKSHMI.J			
GROUP 2					
2.1	920411104004	ABUBUCKER.M	Evolutionary Process Model, Specialized Process Model	Evolutionary Process Model	Evolutionary Pro (22/07/13)
2.2	920411104009	ARAVINDRAJ.R			
2.3	920411104010	ARUN KUMAR.P			
2.4	920411104016	BALASUBRAMANIAN.M			
2.5	920411104058	NAVEEN PRABU.E		Specialized Process Model	Specialized Process M
2.6	920411104063	PRABHAHAR.P			
2.7	920411104076	SAKTHI VIVEKANANDAN.U			
GROUP 3					
3.1	920411104086	SIVA PRASAD.J	System Engineering : Computer Based System, Business	Computer Based System	Computer Based Sys Engineering Hierarchy
3.2	920411104107	VIJAYAKUMAR.R			
3.3	920411104109	VINOTHKUMAR.K			
3.4	920411104304	DHEIVAKARAN.P			

3.5	920411104308	KARUPPASAMY.M	Process Engineering, Product Engineering	Business Process Engineering, Product Engineering	Business Process Engineering, Product Engineering (29/07/13)
3.6	920411104317	SATHISHKUMAR.M			
3.7	920411104325	WILSON.K			
GROUP 4					
4.1	920411104011	ARUNA DEVI.S	Verification, Validation & System Modelling Overview	Verification, Validation	Verification, Validation (01/08/13)
4.2	920411104032	JEYASHREE.B			
4.3	920411104039	KARTHIKA.V			
4.4	920411104048	MEERA.R			
4.5	920411104050	MEKHALA.R		System Modelling Overview	System Modelling Overview (02/08/13)
4.6	920411104055	NANTHINIDEVI.P			
4.7	920411104060	PANDEESWARI.A			
4.8	920411104073	RAMYA JENEFAR GRACE.R			
4.9	920411104080	SHANMUGA PREETI.R			

Experimental Results:

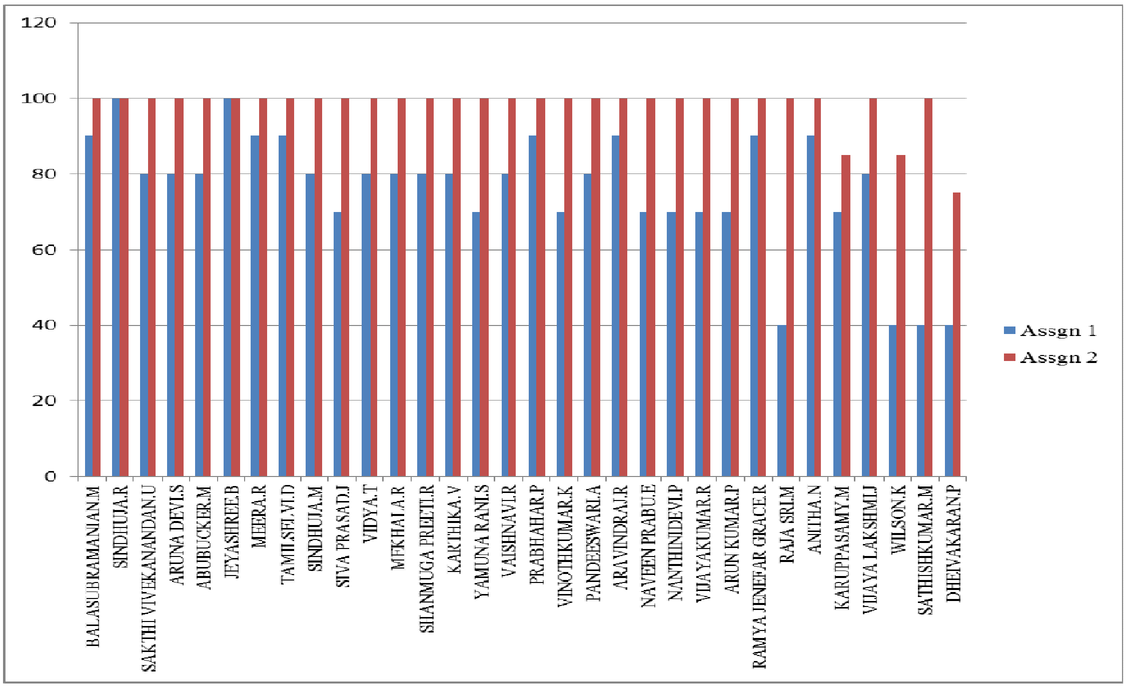

Fig. 2: Assignment Test Analysis

The Assignment Marks were analyzed before and after implementation of the proposed techniques as shown in Fig. 2. The Slip Tests were analyzed as well as in Fig. 3 so as to check the consistency. The Internal Test Marks were analyzed before and after implementation of the proposed pedagogy as shown in Fig. 4.

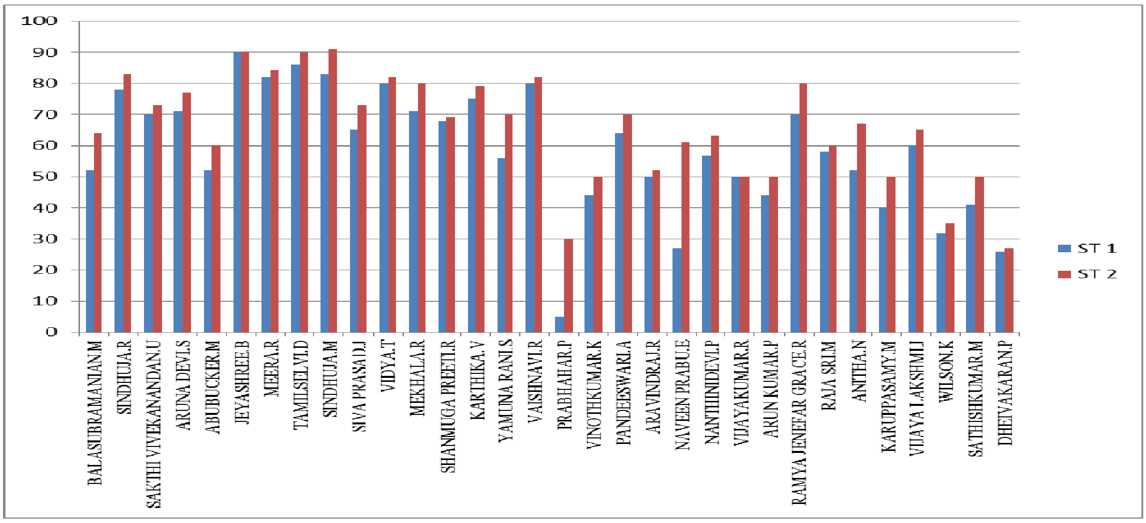

Fig. 3: Slip Test Analysis

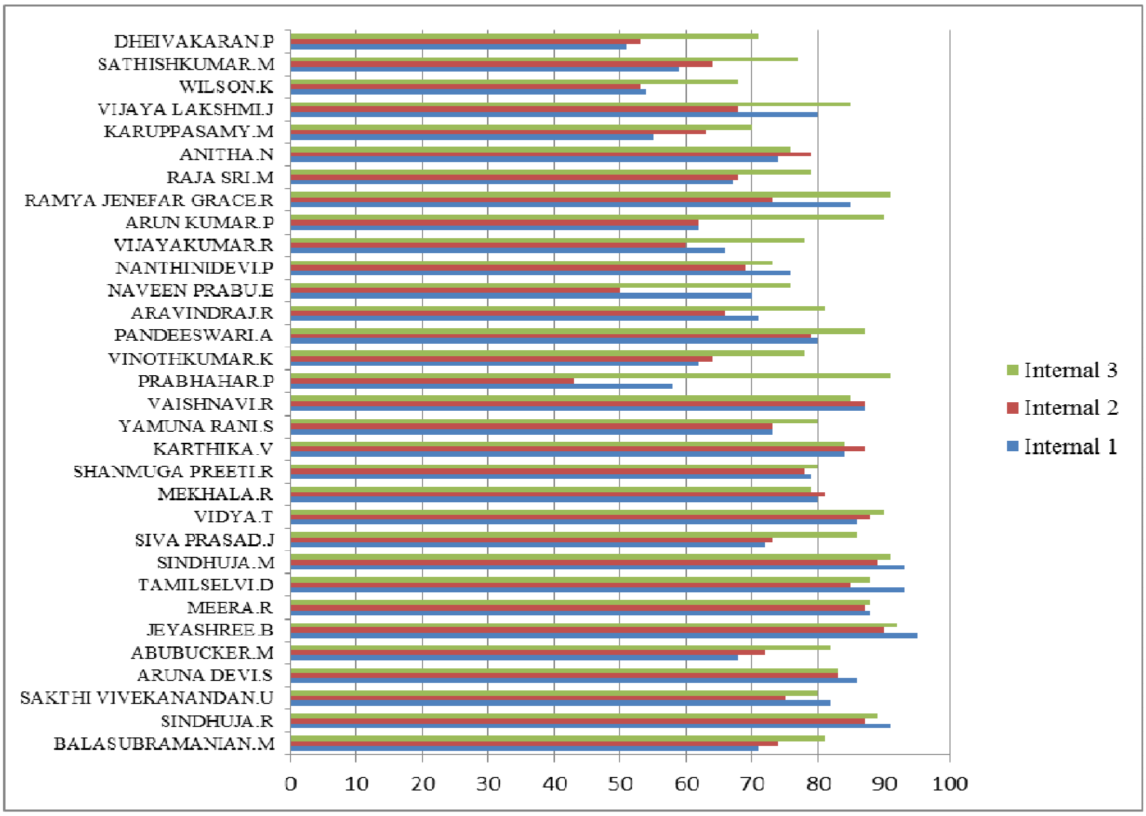

Fig. 4: Internal Test Analysis

The results obtained proved that, the progress of the students were good and hence it made us to extend the same system with some improvisations in other subjects as well. Our aim is to make the students expert in the subjects they learn [4] and to obtain credible outcome in their academic performance.

Conclusion:

The existing systems were compared and after the implementation of the proposed methodology the results indicated that the methodology has been successful. The students were able to think-out-of-the-box and developed eagerness to learn. They were able to improvise and overcome stage-fear. They cultivated the habit of asking questions. This innovative proposed methodology improved the education-based interactions [5] within the student community.

The students came up with fresh ideas and ways of implementing them. They were able to adhere to the deadline. Wide range of materials ensured there are easier methods or materials to study. Leadership and organizational skills were improved.

Future Enhancement:

This is an era of social media and the teaching methodology [8] can only improve with technology. So many technical additions can be done to the proposed methodology- like establishing a social media page. An entry into social media will ensure that this methodology will not be limited to a classroom. E-classrooms can be envisioned [9].

Also there can be forums that can be initiated which will help the system to reach far. More and more students can get access to this methodology and learning materials and it will improve the quality of the students.

REFERENCES

1. Bhuvaneswari, T. and S. Prabaharan, 2013. "*A Survey on Software Life Cycle Models*", International Journal of Computer Science and Mobile Computing, 2(5): 262-267.

2. Apoorva Mishra and Deepti Dubey, 2013. "*A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios*", International Journal of Advance Research in Computer Science and Management Studies, 1(5): 2321-7782.
3. Vanshika Rastogi, 2015. "*Software Development Life Cycle Models – Comparison, Consequences*", International Journal of Computer Science and Information Technologies, 6(1): 168-172.
4. Evrim Baran, 2013. "*Connect, Participate and Learn: Transforming Pedagogies in Higher Education*", Bulletin of the IEEE Technical Committee on Learning Technology, 15: 1.
5. Salah Alkhafaji and B. Sriram, 2012. "*Educational Software Development Life Cycle Stages*", Chinese Business Review, ISSN 1537-1506, 11(1): 128-137.
6. Nayan, B., Ruparelia, 2010. "*Software Development Lifecycle Models*", DOI: 10.1145/1764810.1764814, ACM SIGSOFT Software Engineering Notes, 35: 3.
7. Carol, E., Pollard, Dhiraj Gupta and John W. Satzinger, 2010. "*Teaching Systems Development: A Compelling Case for Integrating the SDLC with the ITSM Lifecycle*", Information Systems Management, 27 (2), DOI: 10.1080/10580531003684959, 113-122.
8. Okamoto, T., 2004. "*Collaborative technology and new e-pedagogy*", IEEE International Conference on Advanced Learning Technologies, 2004, Proceedings, Pages: 1046-1047, ISBN: 0-7695-2181-9, DOI: 10.1109/ICALT.2004.1357746.
9. Isaacs-Sodeye, O. and M. Lande, 2013. "*Teaching with unfamiliar pedagogy for engineering design instructors*", IEEE Frontiers in Education Conference (FIE), ISSN: 0190-5848, DOI: 10.1109/FIE.2013.6685071.
10. Anand, S., S. Chatterjee and K. Bijlani, 2014. "*Pedagogy Experiments with Recorded Video Lecturers*", Technology for Education (T4E), 2014 IEEE Sixth International Conference on, pp: 193-194, DOI: 10.1109/T4E.2014.43.