Journal of Engineering Education Transformations ,
Volume 34 , No. 4, April 2021, ISSN 2349-2473, eISSN 2394-1707

Improved Undergraduate Software Capstone Project Development with Adoption of Industry Practices

Ravindra Mandale[1], Ashwini Patil[2], Amol Adamuthe[3]
[1,2,3]Computer Science & Information Technology Department, RIT, Rajaramnagar.
[1]ravindra.mandale@ritindia.edu
[2]ashwini.patil@ritindia.edu
[3]amol.admuthe@gmail.com

Abstract: To improve cost-effectiveness, the software development processes in the IT industries are changing rapidly. There is a gap in different processes in IT industries and engineering institutes due to a mismatch in industry expectations and academic practices. According to different survey reports, this gap is increasing significantly due to lack of domain knowledge, inabilities of adopting recent technologies, old curriculum contents, poor assessment methodologies, and old project development practices, etc. To make students industry-ready, it is necessary to inculcate recent software development processes in academics. Academic project is one of the important courses in an undergraduate program which inculcates industry required skills among the students to bridge the gap to a greater extent. In this work, the gap in academic software project development practices is identifies through feedback from industry experts, alumni, and previous three years' student projects. This paper presents Industry Oriented Software Engineering Practices (IOSEP) methodology to adopt the recent industry practices in academics for improving students' project quality. The proposed methodology is implemented for the third-year mini project and a final year capstone project for academic year 2018-19. The IOSEP methodology incorporates an agile model, industry coding practices, GitHub platform, modern tools and technologies, testing tools, real-time deployment and, LaTex for documentation. To analyse the effectiveness of the proposed methodology, a new skill-based assessment method and rubrics are designed. K-means clustering is used to analyse students' performance. The Elbow method and silhouette analysis are used to select the number of clusters. Results show that optimal cluster values are three and more than 38% of students are in excellent cluster. The post feedback analysis from faculties and students show that project quality improved compared to previous years on development time, coding practices, technology use, and technical report writing. Industry-sponsored projects, participation in project competitions, student paper publications, and placement statistics are improved than the previous three years.

Keywords: Software development, Agile model, GitHub, LaTex, Employability skills.

Ravindra Mandale
Computer Science & Information Technology Department,
RIT, Rajaramnagar.
ravindra.mandale@ritindia.edu

1. Introduction

According to Aspiring Minds (National Employability Report Engineers, 2019) report, 80 % of Indian engineering graduates are not fit for a job in the current knowledge economy. This report shows that the small changes in the education system will not address the actual problem. Prashant R. Nair (2020) discussed the statistics of National Association of

Software and Service Companies (NASSCOM) in India, only 25% of the fresh graduates are employable in the IT industry. Mandale and Adamuthe (2019), suggested that there is a need for systematic and drastic change to deal with high un-employability numbers. Engineering graduates are not industry-ready due to a lack of employability skills of 2020. The employers across India looking for professional skills, analysing, problem-solving skills, lifelong learning skills, and ethical responsibility in engineering graduates (Blom and Saeki, 2011). There is a wide gap between the IT industry and engineering institutes due to a mismatch in industry expectations and academic practices. This gap is increasing significantly day by day due to lack of domain knowledge, inabilities of adopting recent technologies, old curriculum contents, poor assessment methodologies, etc.

The software industry expects skilled engineering graduates, but unfortunately, students are lagging in communication, problem-solving, critical thinking and coding skills (AICTE Skill report, 2019). A software engineer must work with users to determine their software needs by designing, developing, and testing an application according to the user's specifications. Software engineering course and capstone project are the best opportunity to learn the real-world software development processes. Mary Shaw and Tomayko (1991) presented that group-based learning, practical exposure to theory, understanding client requirements, and software implementation model are essential skills. The recruiters are looking at these skills among the students. Academic projects are expected to prepare the students to be industry-ready by practicing software engineering processes. Project work inspires student proficiencies to go beyond the course knowledge. Student's projects are important and they have various benefits by applying conceptual knowledge to solve real-world problems.

Song et al. (2009), said that the rapid pace of change in the outside environment is convincing these two different worlds to come together to address and solve some of real-world challenges. To satisfy these requirements of the client by developing high quality and cost-effective software products, the IT industries have shifted from traditional approaches to new software engineering practices. Andrew Stratton et al. (1998), summarized different industrial initiatives within the department to improve the industry experience among the students during the graduation period for gaining a better understanding of work culture in the industry and gain real-world skills relevant to software engineering and computer. To keep pace with rapid changes in industrial practice requires changes in the way software engineering is taught (Robert et al., 2017). The authors presented the case study to provide students with relevant skills for industrial software engineering careers, and thus producing graduates who are well versed in modern software engineering principles and practices, and whose skills align well with the needs of industry.

As per Schwalbe, K. (2014), in academics, the waterfall model is used, which is a linear, rigid, and time-consuming model. As per interaction with faculties of different academic institutes, it is found that during the academic projects, the students are using a traditional project development approach, which leads to a technology and implementation gap between industry and academics. By referring to feedback from the Training and Placement Officer (TPO) of 10 different colleges and 25 industry persons, it is found that we are lagging to follow industry-oriented project development practices, which may lead to a poor student employability ratio. To incorporate industry requirements, there is a need to change academic practices to inculcate the required skill among the students during their engineering education. Current software development practices in the industry, project development in academics and industry, and academic are different. Both have different processes and approaches. Considering the growing nature of industry and changes in the project development processes, it has become important to establish how best the academic projects can address various requirements and demands of the dynamic industry.

The main objective of the paper is to provide an industry-oriented capstone software development methodology to undergraduate students. The aim is to minimize the gap between software deployment practices in industry and academics.

The important contributions of the paper are as follows.

- The feedbacks from IT industry experts are taken. The analysis shows that there is a need to replace the existing model with advanced models such as Agile. A customized agile based software development approach for academic projects is presented in this paper.

- While integrating the different modules as a whole project, the students face many difficulties such as naming convention ambiguity, package conflict, etc. which can be resolved by using code integration platforms.

- Most of the time, students prefer a manual testing approach for testing their applications which is not an efficient way to do it. Many automated testing tools like Selenium are used by IT industries. Students should also be familiar with these tools.

The primary goal of our research was to adopt recent software development practices, tools, and technologies in academics to improve the employability skills among engineering undergraduate students. The research design/methods used for our work are given in table 1.

Table 1 : Research Design and Methods

Sub Question	Research design/method used
To identify the gap in industry and academic practices for undergraduate software capstone project development	• Previous years project report analysis by industry expert • Alumni feedback for ongoing projects • Industry expert feedback
To identify the limitations of existing project devolvement practices in academics	• Feedback from academicians and training & placement officers from different institutes • Fishbone diagram (root cause analysis)
To design and develop a methodology to adopt Industry-Oriented practices in software project development	• WHY-WHY analysis • Proposed new methodology
To evaluate students' performance based on project evaluation parameters	• New proposed rubrics • Analysis of students evaluation marks using K-means clustering algorithm

The remaining paper is divided into various sections. Section 2 is about related work, Section 3 presents the methodology used to identify the gap between the industry and academics practices, the proposed methodology is discussed in Section 4, Section 5 discusses the result and the conclusion is given in Section 6.

2. Related Work

A. Outcome-based practices for a project course

The National Board of Accreditation (NBA) in India has come up with qualities that engineering graduates has to demonstrate which are called graduate attributes (GAs').

Following are the engineering GAs':

- Problem analysis
- Design and development of solutions
- Modern tool usage
- Ethics
- Individual and team-work
- Communication
- Project management
- Life-long learning

GAs' are important factors in planning the curriculum of any undergraduate programme. In the Outcome-Based Education (OBE) framework, the Program Outcomes (POs) should be defined as per GAs. The POs are attained through the course outcome (COs). As per Norazlan and Hadzli (2010), these outcomes include a range of knowledge, skills, and attitudes. Project is one of those courses which contributed more in achieving POs. Roberts, P and et al. (2015), which are aimed so that the projects course involves applying the knowledge, and skills from the subjects learned by students through interactions with a real client. At the undergraduate engineering level, the capstone project in the final year & mini project in the third year is very important. The main objective of the project course in the curriculum is to apply the integrated knowledge of subjects learned by students in previous years to solve real-world problems. Project related courses provide an opportunity for assessment of program outcomes. In the capstone project course, the expected thing is to provide the students with informal training on the key elements of project management. Few important points are listed below.

- Solve real-world problems using knowledge of engineering, maths, science.
- Analyze the given problem
- Design and develop the solution for the given problem

- Use modern tools
- Individual and team-work
- Communication
- Lifelong learning
- Time Management
- Research Planning

Thayer and Endres (1987), emphases on the project course because it will help the student to make the transition from a learning environment to the professional environment. This course will help to provide practice of the software development approach during the graduation period to bridge the gap between the industry and academics. Robillard & Robillard (1998) give the relationship between projects in academic and industry and suggest seven recommendations to bring academic projects closer to industrial projects, including basic training in management skills, reducing the relative effort required to acquire the knowledge of the application domain, reducing the urge to code early, testing activities, importance, and usefulness of appropriate documentation, practices involved in the technical review meeting and the roles of a teammate. MacKrell, D (2010) proposed that the employability of the students can be enhanced through a project in which workplace organizational and cross-cultural awareness is embedded with an expected outcome to enhance the relationship with industry, preparing work ready graduates with professional, lifelong learning skills. Keller et al. (2011) suggest that employability skills can be enhanced through involving clients in software development projects. Real client model project helps to improve the employability skills of the students.

Włodarski and Poniszewska (2017) discussed three parameters such as project quality, project efficiency with social factors and client satisfaction for judging the academic project. Posthuma et al., (2019) said that software engineering covers tools, techniques, and the skills for team collaboration and competencies and it gives a unique opportunity for students to get hands-on-experience of software development skills needed by IT industries. The goal of the project course is an opportunity for students to work on one fairly large software project as a small group. Skruch et al. (2017) said that the project exposes the students to practical problems that are crucial in industry engineering problems. While mentoring the project, faculty supervisors should create the environment to give maximum opportunities for the students to learn as per the need of the industry.

Thayer and Endres(1987), recommended a project course that helps the student to make the transition from the learning environment to the professional environment. This course will help to practice the software development approach during the graduation period to bridge the gap between the industry and academics. The project plays a very important role in testing the knowledge of students in all aspects such as domain knowledge, soft skills, team management skills, communication skills, etc. The student projects are considered as genuine proof of the technical expertise of the student during their engineering education.

B. Existing project development practice

At RIT, it is observed that students are using a waterfall model for the development of their academic projects. Students initially collect the requirements from the client. After collecting the requirements, they design and implement the project. In this development model, the project is tested at the end, so not able to incorporate the changes demanded by the client during implementation.

Students are working in a team by dividing the project task into different modules and individual students work on each module. Students face challenges while integrating the different modules as a whole project such as convention ambiguity, package conflict, etc. which may resolve by using code integration platforms. Most of the time, student prefers a manual testing approach for testing their applications which is not an efficient way to test the project. As a part of project report preparation, students are using Microsoft Word for documentation of their project.

The previous evaluations for projects in the IT department focus more on technical requirements of projects such as tools, technologies, coding done, reports, etc. It loses its focus on teamwork, communication, software engineering practices used, latest technologies used in the IT industry for project activities. Along with this, only faculty mentors were assigned to project groups as guides for the project development.

3. Methodology to identify the gap in industry and academic practices

The analysis to identify the technology and implementation gap between academics and industry is completed by taking three feedbacks.

- Previous years project report analysis by industry expert
- Alumni feedback on usage of modern tools and technologies for ongoing projects
- Industry expert feedback for the adoption of tools and technologies in project development.

A. Previous years project report analysis

Analysis of previous three years' projects such as 2015-16, 2016-17, and 2018-19 was done using their reports. Project reports were sent for review to an industry expert. The analysis of these reports was done using following the questionaries.

- Whether the software development life cycle (SDLC) was followed or not?
- Whether requirement analysis and design are as per industry standards or not?
- Whether implementation and testing are done using industry standards or not?
- Whether the report follows the guidelines of technical report writing or not?
- Whether technology used for projects is appropriate or not?
- Whether project work is innovative or not?
- Whether project deliverables are satisfactory or not?

After analysis of feedbacks given by industry experts, the following lacunas are identified in current academic project development practices.

- Students are using the waterfall model for project development.
- They are not using appropriate tools, techniques, and coding standards.
- They are not using any appropriate code sharing/ integration platform.
- Students are doing manual testing for the project.
- They are using Microsoft Word for documentation.

B. Alumni and industry expert feedback analysis

The second survey was conducted to understand the recent industry practices for software development. The feedback from 26 different IT companies are taken regarding industry software development practices. Table 2 presents feedbacks taken from industry alumni and experts regarding industry software development practices. After analysing the feedback received from the industry, it is found that the students are not using software development practices that are adopted by industries. Current industry practices, tools used for project implementation, a comprehensive list of testing tools, and report writing tools used in the industry are received through these feedbacks. In addition to this, the pitfall of our previous students' projects is obtained from these feedbacks.

The following conclusions were made after analysis of the feedback:

- The old software development process (waterfall model) is not efficient to manage the rapid change in the requirements of the customer. To satisfy the changes of the customer requirement in a short period and to deliver the product in an efficient manner, industries have adopted the agile model for software development which proves for their effectiveness and is transforming the software industry. Most of the IT industries are using the agile model for software development. Whereas, for academic student projects waterfall model is used.

  Industry experts agreed that the software engineering model followed in academics needs to change. Almost 81% of industry experts say that there is a need to change the software practices.

- The student uses a manual approach for code sharing and integration during their graduation. IT industries are using various modern tools for code integration and sharing. GitHub is majorly used for code integration and sharing.

- Industries are using automatic testing tools like Selenium, JMeter, etc.
- For documentation purposes, industries are using tools like Jira, Confluence, etc.

Table 2 : Feedbacks on industry practices

Que. No	Question	Response	Percentage
1	Is the project development process in academics need to change?	Agree	80.8
		Neutral	11.5
		Disagree	7.7
2	Which SDLC model is used by your organization?	Agile	96
		Code and Fix	1
		Waterfall	2
		Iterative	1
		Incremental	0
		Spiral	0
		Prototyping	0
3	How did you integrate code for your final year project?	Manually	35
		Git	25
		Direct	8
		Nil-ED	8
		Modules	8
		NetBean editor	8
		TFS	8
4	Which platform are you using currently for code integration?	Linux	4
		Visual Studio	4
		GitLab	11
		Jenkin	4
		BitBucket	11
		Git	50
		Microsoft Team foundation Server	4
		TFS	7
		Salesforce	4
		Android studio	1
5	Which testing tools are you using in your organization for testing projects such as web applications, mobile apps?	Selenium	48
		Manual	8
		Jenkins	4
		Protractor	7
		JMeter	11
		Karma	4
		Cucumber	4
		Powershell	4
		Ranorex	4
		Appium	4
		SonarCub	2
6	Which tools are you using in your industry for documentation or report writing?	Jira	19
		Confluence	33
		GoogleDoc	10
		MS- Office	19
		AGM	10
		TFS	5
		Read the Docs	5

Following are the skills or knowledge the software industries are expected from students of IT professional:

- Knowledge of the latest SDLC model used in the industry for software development.
- Knowledge of the latest tool for integration and sharing the code.
- Knowledge of tools for documentation and report writing.

Due to the lack of above skills in students, IT industries are spending time and money on training. Nowadays, industries are looking for industry-ready students. Therefore, it is necessary to bridge the gap in academics and industry by incorporating the skills required by industries among the students during their course of study.

C. Limitations of existing project devolvement practices in academics

The gap in software development practices in industry and academics are identified through the two surveys conducted. The causes behind the poor-quality projects and weak student's skillset are identified based on the survey analysis. The gap is represented using a cause and effect diagram. Fig. 3 presents the Ishikawa diagram (fishbone diagram) for the problem. The oval symbol under each category indicates the causes that are found for the project development. Ishikawa diagram (fishbone diagram) is a popular cause and effect diagram. It is used in various applications to identify the root causes. E.g. Ishikawa diagram is used for identification of causes in project management by Sheng-lei, (2009), operation management by Luo et al. (2007), analysing diagnostic errors by Reilly et al. (2014) and identify the staff and student-related problems in technical education by Shinde et al. (2018).

The major causes are explained below.

- Project development model

The waterfall software development model is linear, rigid, and time-consuming. This model is not suitable for projects where the client requirements are changing from time to time. There is less involvement of clients in the project, and less client satisfaction ratio.

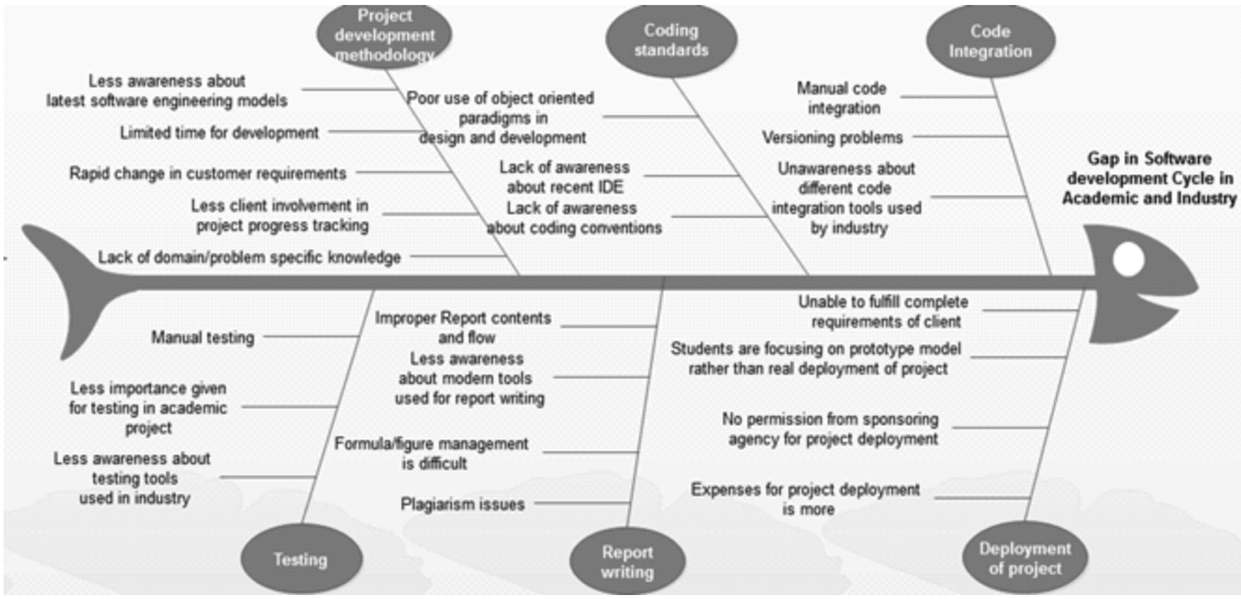


Fig. 1 : Fishbone diagram

- Coding standards
- Code integration

Manual code integration is used for different modules of projects. It is difficult for team members to work on files and easily merge their changes with the root source code of the project while integrating. It creates naming convention ambiguity and packages conflict problems.

- Testing

Students are doing only manual testing.

- Report writing

Using Ms-Word, it's difficult to handle equations, figures, bibliographies, indexes, etc.

- Project deployment

A WHY-WHY analysis is applied to the overall causes identified during the problem analysis. Fig. 2 shows WHY-WHY analysis which is performed through brainstorming by the team members, with the assistance of the head of the department, faculty members, Training and Placement Coordinator(TPC) of the department. To increase the students' project quality, there is a need to improve the strategy currently adopted by the department. After doing WHY-WHY analysis, the solution i.e. "Adoption of new Industry Oriented Software Engineering Practices for improving students projects" was proposed.

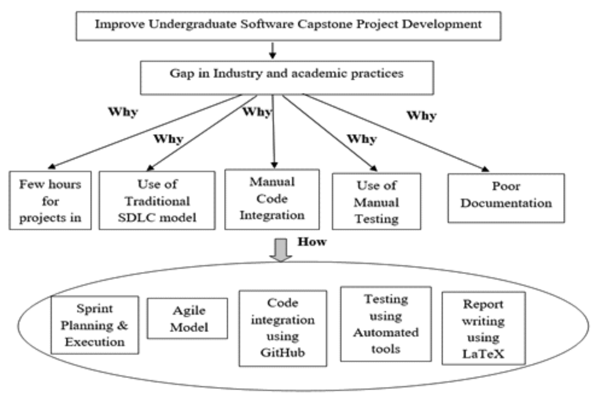


Fig. 2 : WHY-WHY analysis

4. Proposed IOSEP Methodology

This section presents the IOSEP methodology for improving the software project quality in academics. This methodology consists of the following 6 phases. Fig. 3 shows the proposed IOSEP methodology.

The implementation of the proposed methodology is done by changing the project development practices, to incorporate the industry required skills among the students. The major changes in conventional practices are,

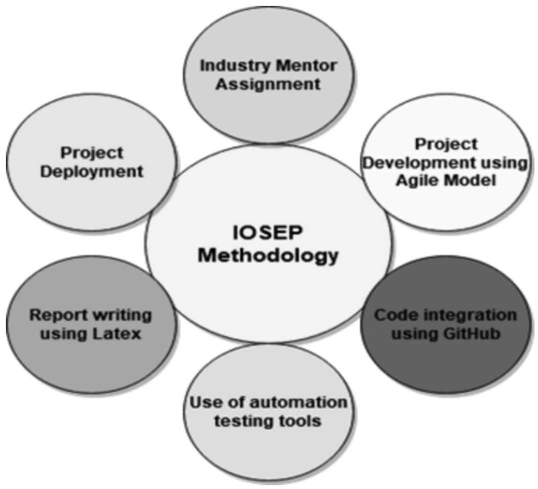


Fig. 3 : IOSEP Methodology

- Assigned industry mentor for each project group
- Adopted agile model for software development instead of waterfall model
- Use of automated testing tool instead of manual testing
- Document such as synopsis, SRS, project report are prepared using LaTex instead of using MS-Word
- Deployment of the project on different platforms
- Use of GitHub for code Integration and sharing

A. Industry mentor assignment

The industry mentor can guide student projects as per industry standards such as technology and tools selection for projects. This will expose students to industry culture in the college campus only. Once the project topics are finalized by college mentors with panel members, the projects are categorized as per the domains. Industry mentors are identified by the department Training and Placement Coordinator (TPC) and Alumni coordinator. The finalized topics are assigned to each industry mentor to guide and monitor activities as per the industry perspectives. Faculty and alumni collaboratively guide students for their project work.

B. Project development with the agile model

SDLC is a systematic process for developing high-quality software products that satisfy the clients by fulfilling their needs and expectations. This SDLC consists of several phases: Requirement gathering and analysis, design, coding, testing, real-time deployment, and maintenance. There are many SDLC models to carry out this project development process. These models include the waterfall model, spiral, prototyping, iterative development approach, agile, etc. In academics, the software projects are carried out using a waterfall model which has some limitations. As per our survey, currently to speed up the concept-to-commercialization process, Agile model is the best model for product development. Agile model is a more collaborative approach. Project work is incremental and the focus is on rapid delivery. Rajeshwari et al. (2016), suggested to inculcate the concepts from Agile models into students in an earlier stage to make them industry-ready. Vijayalakshmi et al. (2018) highlight the need for transition from a traditional software process model to agile model for the implementation of a project in academics. They used the agile model for project implementation in academics and proved the effectiveness of agile model for academic projects. Authors also suggested designing the curriculum in collaboration with the industry for incorporating industry practices in academics which is required to train the students to face the real world. Jayawardena and Ekanayake (2010) said that the client gets involved throughout the project development stages. Agile model encourages continuous rehearsal of development and testing throughout the software development phases of the project. Both development and testing activities are parallel, unlike the waterfall model. Here, various terminologies of Agile model are used such as sprint, scrum, scrum master, project owner, etc. Each group is advised to prepare a sprint for each phase of SDLC to split the activities and speed the project work. Agile models have the following five components:

i) Product roadmap creation

The entire project implementation is broken down into modules to create the sprint planning for each module separately.

ii) Release planning

The shorter development cycles (called sprints) are considered to release each module in a short period.

iii) Sprint planning

In every module, prepare a to-do-list along with a deadline. The task is eventually divided among team members so they can accomplish their assigned tasks during the sprint as per the deadline.

iv) Weekly twice meetings planned with guide

During these meetings, each team member will briefly talk about what is done and what is planned next.

v) Sprint review and retrospective

After the end of each sprint, student's meetings are kept with a guide to show the demo of the module. The results of each module were discussed with industry mentors.

C. Code integration using GitHub

In the traditional approach, students work in a team, they develop code separately for the different modules. The traditional way of integrating the code of different modules faces many difficulties such as naming convention ambiguity, package conflict, etc. These problems can be resolved by using GitHub. This platform allows us to integrate code in a distributed environment as well as deployment into a shared repository. GitHub also provides facilities for contributing to the open-source community. Raibulet and Fontana (2018), introduced the use of GitHub, Microsoft tool for evaluating collaboration and teamwork.

D. Testing using automated tools

Various automation testing tools are used in industries such as Selenium, Junit, etc. To get familiar with such tools, students are encouraged to prefer these along with manual testing. For mobile applications using Android, encouraged students for Junit framework, Monkey, device testing, as there are multiple screen size devices available with users. This is a very important type of testing as far as mobile apps are concerned. For mobile and web applications, validation and verification testing is suggested.

E. Project deployment

As per the type of project, the deployment platform varies, e.g. mobile application requires Google Play-Store, web application requires domain for hosting. To make the students familiar with such platforms, different sessions are conducted sessions. So that the 100 % project deployment can be achieved in a real-time environment using Google Play, by purchasing a domain (If required), client location in case of sponsored projects, etc. If students would like to contribute to open-source communities, they can share their code on GitHub after deployment.

F. Report writing using LaTex

The students are encouraged to use various document preparation systems like Latex, Jira, and Confluence, etc. which are standard tools for report preparation. As far as the academic institutes are concerned students need to publish the paper on their project/research work. As most of the conferences, journals, and proceedings, ask authors to submit their papers in LaTex format. Therefore, use of the LaTex tool is recommended for report writing. The final reports are checked using plagiarism software.

Role and responsibilities of students and faculty mentors during the implementation of IOSEP methodology:

The project exposes the students to practical problems that are crucial in industry problems. Skruch et al. (2017) suggest that, while mentoring the project, faculty supervisors should create the environment to give maximum opportunities for the students to learn as per the need of the industry.

Roles and responsibilities of students

- To Prepare Sprint documents for project planning and execution
- To conduct daily meetings with scrum
- Regular discussion with mentors
- Use of project diary
- Follow guidelines given by the project coordinator
- Complete project activities are stipulated time

Roles and responsibilities of institute mentors

- Interaction with an industry mentor
- Guide project group in project activity

- Take stand up meeting twice a week to discuss the issue and progress
- Motivate students to make use of the latest tools, technologies, frameworks
- Motivate students to use the IOSEP methodology for project development.

5. Results and Discussion

The result analysis and discussions are presented in this section. The IOSEP methodology is applied to third-year and final year projects. There are 16 mini projects and 12 final year B. Tech projects. A total 120 students were involved in implementation of the IOSEP methodology. Derrick and Stevens (2005), proposed the award competition for best student projects for improving students' engagement in project activity to improve the self-learning abilities, confidence, and modern tool usage qualities. Sánchez et al. (2014) presented evaluation and assessment criteria for testing student's professional skills such as entrepreneurial attitude, innovation, sustainability, and social commitment, foreign language awareness, effective oral and written communication, teamwork, proper use of information resources in project work. They presented a proposal for the evaluation of the final year project based on the recommendations of “Guidelines to the evaluation of competencies in the Bachelor and Master Degree thesis in engineering”. The assessment is based on three evaluation criteria like an initial milestone, follow-up milestone, and a final milestone. Boettger (2010), has proposed an assessment tool with rubrics for the technical communication of students. The author found the limitations of traditional evaluation such as bias, whims. Rubrics play a vital role in the indirect measurement of students. Rubrics help evaluators for transparent evaluations and clearly distinguishes the students. They are strongly recommended for project work, as we cannot measure soft skills like work using direct measurement. Mandale and Adamuthe, (2019), applied the skill-based evaluation technique for evaluating the third year and final year project by considering project scope, deadlines, and technologies. For final year B.Tech projects, the performance is judged using parameters like the number of sponsored projects, participation in project competition, and paper publication, GitHub account, its usage, and number of reports prepared using LaTex. There are total 8 sponsored projects out of 12 projects. Out of 12 groups, 3 groups have participated in project competition held outside the college, 1 group won the prize. All 12 groups have participated in college-level project competition i.e. Quantum, 3 groups won prizes. A total 3 papers are accepted in UGC approved journals. All 12 groups deployed their projects on GitHub. All 12 groups prepared their final year B.Tech project reports using the LaTex tool with plagiarism and grammar checking. The overall comparisons of improvement for project achievements are given in Fig. 4. Looking at the statistical analysis of data for the academic year 2017-18 and 2018-19, there are significant improvements in the industry-sponsored projects, participation in outside project competition, paper publication, and placement.

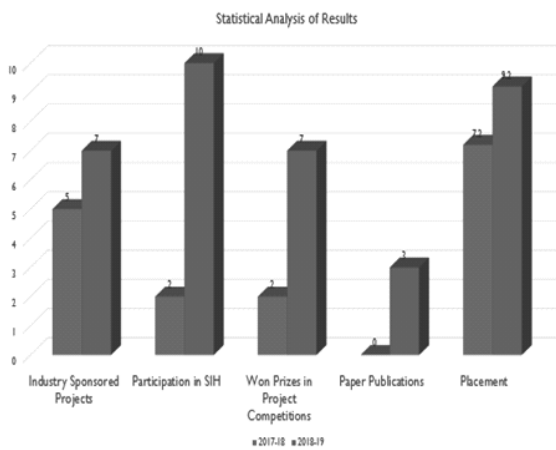


Fig. 4 : Comparison of improvement

For the third year, the performance is checked based on parameters namely participation in competitions, GitHub account, its usage, and the number of reports prepared using Latex. Out of 16 groups, 7 groups have participated in a college level Hackathon and 3 groups won prizes. In the National level Hackathon i.e. Smart India Hackathon, total 10 groups are participated and received appreciation from panel members. All 16 groups deployed their projects on GitHub. All 16 groups prepared their mini-project reports using the LaTex tool with plagiarism and grammar checking.

The project evaluation criteria for the academic year 2017-18 are as follows,

- Requirement specifications for all the modules
- Analysis and evaluation of information

- Implementation/Coding of the project
- Innovativeness in the proposed solution
- Use of coding conventions
- Understanding and contribution in coding the modules
- Testing of the project and use of appropriate tools/techniques
- Sponsored/in-house project
- Participation in the project competition
- Research paper publication
- Quality of project report and timely submission

The evaluation is done as per rubrics with a focus on all aspects of the why-why analysis and IOSEP methodology. The modified project evaluation criteria for the academic year 2018-19 are as follows.

- Adoption of the latest software engineering practices used in industries (Agile/Code and Fix /Prototyping etc.) for development
- Use of appropriate tools/technologies for coding the modules, use of coding conventions
- Project integration using GitHub
- Testing of the project with appropriate use of tools/techniques
- Project deployment using GitHub/Google Play/client-side
- Use of report writing (LaTex), English checking, and Plagiarism tools
- Project work completion as plan as per sprint planning
- Participation in the project competition
- Research paper publication
- Quality of project report and timely submission

There are 6 parameters considered in the evaluation of project marks namely project development with Agile model (P1), implementation using modern tools/techniques and coding practices(P2), code integration using GitHub (P3), testing using automated tools (P4), report writing using LaTex (P5), project deployment (P6). Fig. 5 shows the results of evaluation marks of the entire final year class on a normalized score of 0 (lowest) to 5 (highest). Median values for all parameters are more than 4. Students have shown better adoption of coding conventions, code integration & deployment tools, testing tools, and report writing. There is scope to improve the adoption of agile software development model.

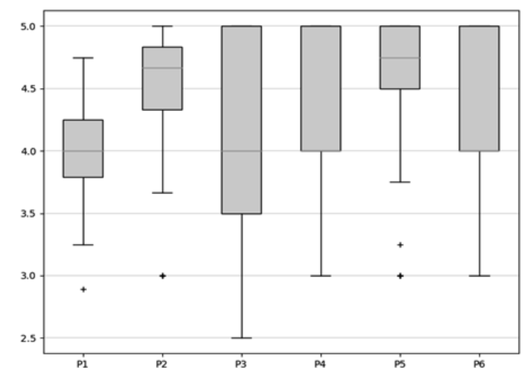


Fig.5: Evaluation of project marks for the year 2018-19

Oyelade et al. (2010) used the clustering algorithm and it is found a good benchmark to check the student's progression of marks. The clustering is a technique used to divide the population into number of groups called clusters. Similar items are grouped in a cluster. In this work, students are clustered based on marks obtained in mini-project and final year B. Tech projects. In this paper students are clustered into groups such as excellent performers, good performers average performers and low performers. K-means, Mean-Shift Clustering, DBSCAN and EM using GMM are popular clustering Algorithms. K-means algorithm is the simplest unsupervised clustering algorithm. K-means clustering is easy to apply. Many applications from different domains such as marketing, image processing, information retrieval, market research are investigated using K-means clustering algorithm. It is used in data mining and pattern recognition fields (Li, Youguo & Wu, Haiyan, 2012). The experimentation of K-means algorithm is done with the help of the Weka tool and python programming language with supporting libraries. The

Weka is graphical use interface (GUI) based tool consisting of various machine learning algorithms for data mining tasks. The python is free and open source, general-purpose programming language. The python is well known for its set of libraries used in the fields like machine learning, web development, natural language processing, etc. Table 3 shows different analysis measures used for Weka tool.

Table 3 : Analysis measures

Settings	Values
Sample size	119
Attributes	6 (P1 to P6)
Minimum value	2.889
Maximum value	4.75
Mean	3.997
No. of clusters	3
Standard Deviation	0.372

The result of the pre-processing phase is presented with the help of a Histogram as shown in Fig. 6.

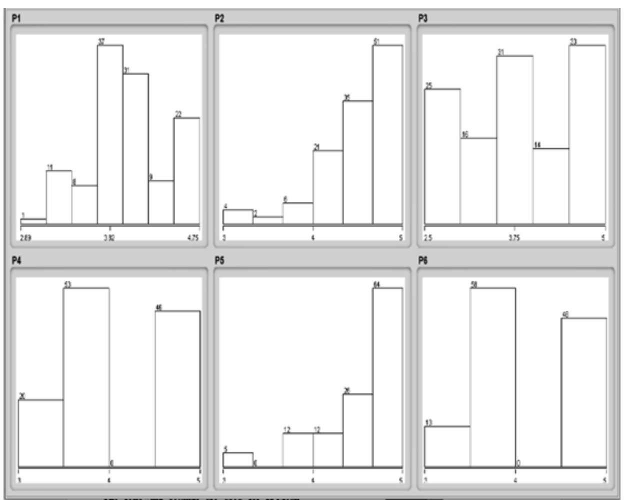


Fig. 6 : Histogram for all parameters

It is important task to determine the optimal number of clusters for the sample problem instance. The Elbow is one of the most popular methods to determine optimal cluster value. Fig. 7 shows the results of Elbow method for given problem. To determine the optimal number of clusters, we have to select the value of k at the “elbow” i.e., the point after which the distortion/inertia starts decreasing in a linear fashion. Thus, for the given data set, the optimal number of clusters are three.

For further analysis, the Silhouette analysis is used to choose an optimal value for numbers of clusters. Silhouette analysis is used to study the separation

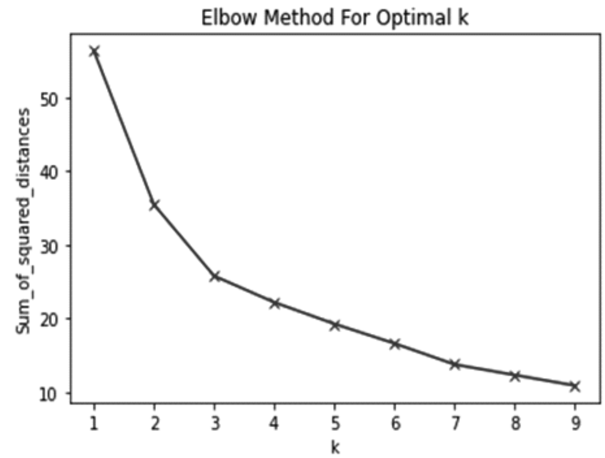


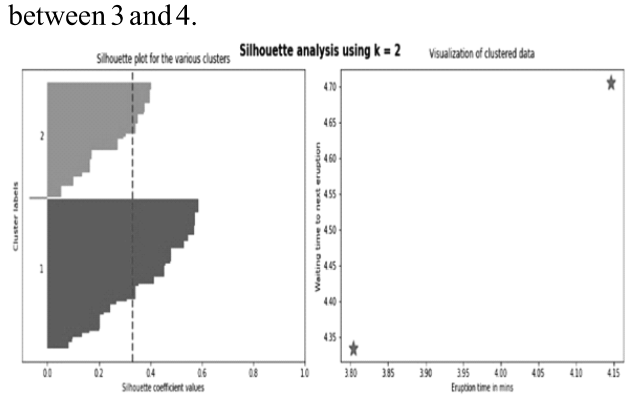
Fig. 7 : Elbow method for deciding no. of clusters

distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighbouring clusters and thus provides a way to assess parameters like the number of clusters visually. This measure has a range of [-1, 1]. The Silhouette coefficient for a different numbers of clusters is given in table 4. Silhouette coefficients (as these values are referred to as) near +1 indicate that the sample is far away from the neighbouring clusters.

Table 4 : Silhouette coefficient

Numbers of clusters	Score
2	0.330
3	0.346
4	0.364
5	0.362

The silhouette plot shown in Fig. 8 indicates that 4 and 5 number of clusters are a wrong choice for the given data due to the presence of clusters with below-average silhouette scores and also due to wide fluctuations in the size of the silhouette plots. Silhouette analysis is more ambivalent in deciding between 3 and 4.



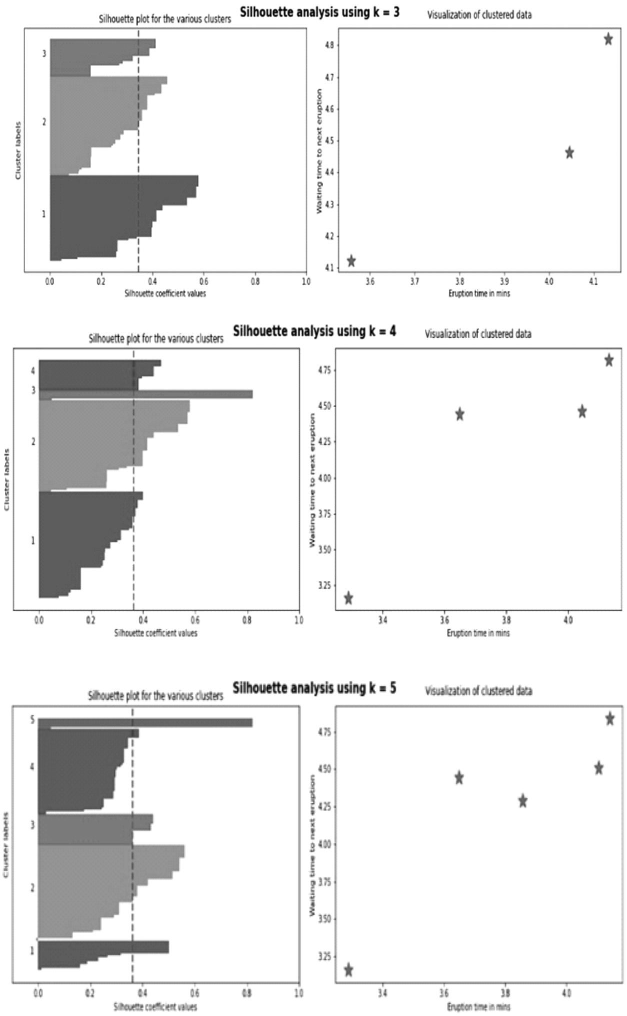


Fig. 8 : Silhouette analysis

Also, from the thickness of the silhouette plot, the cluster size can be visualized. The silhouette plot, when n_clusters is equal to 2, is bigger owing to the grouping of the 3 sub-clusters into one big cluster. However, when the n_clusters is equal to 3, all the plots are more or less of similar thickness and hence 3 is a good choice for numbers of clusters.

Table 5 : Cluster results

Cluster Name	Cluster size
Excellent	46
Good	53
Average	20

Based on the input dataset, settings, and parameters considered in the evaluation of marks, results are obtained and they are shown in Table 5 as cluster results. The majority of students are present on above-average cluster.

The post-activity feedback on implemented practices of the proposed IOSEP methodology from faculties and students are taken. The post-activity feedback parameters for faculties and students are different. The feedback parameters (questionnaire) for faculties are as follows. The responses are Yes and No. The feedback results are indicated using Table 6. The above feedback shows that the IOSEP methodology for academic project development helped students to improve their project quality, report quality, active participation in paper publication, and active participation count in a project competition. This practice improved the knowledge of students regarding the SDLC model i.e. Agile, latest technologies for implementation, code integration and sharing tool, testing tool, and LaTex tool.

Table 6 : Feedback on implemented practices

Que. No.	Questions	Feedback Grades	Feedback Percentage
Mentor feedback			
1	Sprint planning helped in project implementation.	Yes	88.9
		No	11.1
2	Students used GitHub for code integration and sharing.	Yes	100
		No	0
3	Students used industry - accepted testing tool for testing their projects.	Yes	77.8
		No	22.2
4	Students used LaTex tool for preparing the reports	Yes	88.9
		No	11.1
Student Feedback			
1	Agile model helped you for fast project development.	Yes	84.2
		No	15.8
2	GitHub is the best platform for code integration and sharing.	Yes	78.9
		No	21.1
3	LaTex tool improved project report quality.	Yes	63.2
		No	36.8
4	This practice helps you in your placement drives.	Yes	89.5
		No	10.5

6. Conclusions

Adoption of new industry software engineering practices is necessary to make the student's industry-ready with the knowledge of the latest tools, technologies and practices used by IT industries. In this paper, we proposed the IOSEP methodology for academic software project development which helped in improving the overall quality of the project work. The IOSEP methodology is defined based on the feedback analysis of IT industry experts, alumni, and previous three-year projects. The IOSEP methodology is implemented for third-year mini project and final-year capstone project in the academic year 2018-19. The IOSEP methodology achieved better student performance for the year

2018-19 as compared to previous years. This paper presents skill-based evaluation rubrics for measuring the students' performance in terms of use of the latest SDLC model, industry coding practices, code integration platforms, latest testing and report writing tools, getting industry-sponsored projects, participation in project competitions, and paper publication. The result shows that there is significant improvement in industry sponsored projects, participation and achievement in project competitions as well as paper publications. It is also found that the student placement percentage of the year 2018-19 is increased by 20% as compared to 2017-18. In this paper, we used the K-means clustering algorithm for checking students' progress in capstone and mini-projects. The mean values of student's marks are found excellent. The students are clustered in three classes: excellent, good, and average. The majority of students are above average. Results show that the proposed methodology helps to bridge the gap between academics and industry to a greater extent, subsequently increase the employability ratio.

References

[1] AICTE Skill report 2019. https://www.aicte-india.org/sites/default/files/India Skill Report-2019.pdf.

[2] Blom, A., & Saeki, H. (2011). Employability and skill set of newly graduated engineers in India. The World Bank.

[3] Boettger, R. K. (2010). Rubric use in technical communication: Exploring the process of creating valid and reliable assessment tools. IEEE transactions on professional communication, 53(1), 4-17.

[4] Chatley, R., & Field, T. (2017, May). Lean learning-applying lean techniques to improve software engineering education. In 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET) (pp. 117-126). IEEE.

[5] Derrick, E. J., & Stevens, K. T. (2005, October). Work in progress-using emerging technologies to assess student software development projects. In Proceedings Frontiers in Education 35th Annual Conference (pp. F4G-19). IEEE.

[6] Hashim, N., & Hashim, H. (2010, July). Outcome based education performance evaluation on the final year degree project. In the Proceedings of the 7th WSEAS international conference on engineering education, 215-222.

[7] Jayawardena, D. S., & Ekanayake, L. L. (2010, September). Adaptation analysis of agile project management for managing it projects in sri lanka. In 2010 International Conference on Advances in ICT for Emerging Regions (ICTer) (pp. 1-4). IEEE.

[8] Keller, S., Parker, C. M., & Chan, C. (2011). Employability skills: Student perceptions of an IS final year capstone subject. Innovation in Teaching and Learning in Information and Computer Sciences, 10(2), 4-15.

[9] Li, Youguo & Wu, Haiyan. (2012). A Clustering Method Based on K-Means Algorithm. Physics Procedia. 25. 1104-1109. 10.1016/j.phpro.2012.03.206.

[10] Luo, Y. M., Huang, S. Y., & Cao, S. Y. (2007). Application of Improved fishbone diagram in the Operation Management. INDUSTRIAL ENGINEERING JOURNAL-GUANGZOU-, 10(2), 138.

[11] MacKrell, D. (2010, June). A project to improve the employability skills of postgraduate ICT students through workplace cultural awareness. In Proceedings of the Informing Science & IT Education Conference (InSITE), Bari (pp. 17-18).

[12] National Employability Report Engineers (2019), Aspiring Minds, Available online at https://www.aspiringminds.com/research-reports/national-employability-report-for-engineers-2019

[13] Oyelade, O. J., Oladipupo, O. O., & Obagbuwa, I. C. (2010). Application of K-means Clustering algorithm for prediction of Students Academic Performance. arXiv preprint arXiv:1002.2425.

[14] Posthuma, J., Pieterse, V., & Baror, S. (2019, November). A class project to prepare software engineering students for their capstone projects. In Proceedings of the 8th Computer Science Education Research Conference (pp. 66-78).

[15] Prashant R. Nair (2020), Increasing Employability of Indian Engineering Graduates through Experiential Learning Programs and Competitive Programming: Case Study, Procedia Computer Science, 172, 831-837.

[16] Rafal Włodarski and Aneta Poniszewska-Marańda (2017, September). Measuring dimensions of software engineering projects' success in academic context. In 2017 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 1207-1210). IEEE.

[17] Raibulet, C., & Fontana, F. A. (2018). Collaborative and teamwork software development in an undergraduate software engineering course. Journal of Systems and Software, 144, 409-422.

[18] Rajeshwari, K., Naarayan, B. K., & Babu, K. M. (2016). Android MOOC, Blended MOOC Intertwined with Curriculum for Outcomes-Based Education. Journal of Engineering Education Transformations, 30(1), 68-72.

[19] Ravindra J. Mandale, Amol C. Adamuthe (2019), Skill Based Evaluation Technique for Software Development Projects in Higher Education, National Conference on Exploring New Dimensions in Teaching Learning for Quality Education.

[20] Reilly, J. B., Myers, J. S., Salvador, D., & Trowbridge, R. L. (2014). Use of a novel, modified fishbone diagram to analyze diagnostic errors. Diagnosis, 1(2), 167-171.

[21] Roberts, P., Modi, S., Roubert, F., Simeonova, B., & Stefanidis, A. (2015, September). Information Systems undergraduate degree project: gaining a better understanding of the final year project module. In EuroSymposium on Systems Analysis and Design (pp. 145-170). Springer, Cham.

[22] Robillard, P. N., & Robillard, M. (1998). Improving academic software engineering projects: A comparative study of academic and industry projects. Annals of Software Engineering, 6(1-4), 343-363.

[23] Sánchez, F., Climent, J., Corbalán, J., Fonseca, P., García, J., Herrero, J. R., ... & Cabré, J. (2014, October). Evaluation and assessment of professional skills in the Final Year Project. In 2014 IEEE Frontiers in Education Conference (FIE) Proceedings (pp. 1-8). IEEE.

[24] Schwalbe K. (2014), Information Technology Project Management, 7th edition, Independence: Cengage Learning, page 346.

[25] Shaw, M., & Tomayko, J. E. (1991, October). Models for undergraduate project courses in software engineering. In SEI Conference on Software Engineering Education (pp. 33-71). Springer, Berlin, Heidelberg.

[26] Sheng-lei, P. I. (2009). Research on the PRS Integrated Fishbone Diagram and Its Application in Project Management [J]. China Soft Science, 4, 92-97.

[27] Shinde, D. D., Ahirrao, S., & Prasad, R. (2018). Fishbone diagram: application to identify the root causes of student–staff problems in technical education. Wireless personal communications, 100(2), 653-664.

[28] Skruch, P., Długosz, M., & Mitkowski, W. (2017, June). Improving the success rate of student software projects through developing effort estimation practices. In Polish Control Conference (pp. 869-880). Springer, Cham.

[29] Song, X., Jingchun, F., & Ming, L. (2009, October). Research on IT project life cycle. In 2009 Second International Conference on Intelligent Computation Technology and Automation (Vol. 4, pp. 244-247). IEEE.

[30] Stratton, A., Holcombe, M., & Croll, P. (1998). Improving the quality of software engineering courses through university based industrial projects. In Projects in the computing curriculum (pp. 47-69). Springer, London.

[31] Thayer, R. H., & Endres, L. A. (1987). Software engineering project laboratory: the bridge between university and industry. In Software Engineering Education (pp. 263-291). Springer, New York, NY.

[32] Vijayalakshmi, M., Bhandiwad, A., Mellikeri, K., & Nagesh, P. (2018). Transition from

Conventional to Agile Process Model an Experience Report. Journal of Engineering Education Transformations.

[33]Web references www.github.com, last accessed on 22 Jan 2021.https://www.cs.waikato.ac.nz /ml/weka/, last accessed on 22 Jan 2021. https://www.python.org/, last accessed on 22 Jan 2021

[34] Web references

www.github.com, last accessed on 22 Jan 2021.

https://www.cs.waikato.ac.nz/ml/weka/, last accessed on 22 Jan 2021.

https://www.python.org/, last accessed on 22 Jan 2021.