GLOBAL ACADEMIC RESEARCH INSTITUTE

COLOMBO, SRI LANKA

International Journal of Engineering Science

ISSN 2424-645X

Volume: 02 | Issue: 01

On 30th June 2020

http://www.research.lk

Author: Dr. Rasika Aponsu
IIC University of Technology, Cambodia

GARI Publisher | Software Engineering | Volume: 02 | Issue: 01
Article ID: IN/GARI/JOU/2020/95 | Pages: 27-31 (05)
ISSN 2424-645X | Edit: GARI Editorial Team
Received: 14.04.2020 | Published: 30.06.2020

PROJECT-BASED SOFTWARE ENGINEERING COURSE USING ESSENCE

Dr. Rasika Aponsu

IIC University of Technology

Cambodia

ABSTRACT

Among the most challenging aspects of training software engineers is helping them develop both the technical and soft skills necessary for the job, such as the ability to work in a team, manage time and resources effectively, and communicate effectively with all parties involved in a project. A few of schools, regrettably, do not include project management in their software engineering curricula or simply cover the theoretical foundations. This article details our time spent instructing software engineering in a business-style setting via the lens of a project-based curriculum. Students will use Essence to integrate what they've learned in class with real-world software engineering project experience.

Keywords: Essence, SEMAT, Project based education, Software development processes

INTRODUCTION

Since its inception in 2009, the SEMAT (Software Engineering Method and Theory) initiative [1] has prioritised the integration of innovative concepts in software engineering practices and methods into the education of computer science (CS) professionals. To disseminate the SEMAT ideas among software developers, it is imperative to incorporate the fundamental components of the Essence [2] standard into the curriculum of computer science undergraduate programmes. A specialised teaching programme was subsequently initiated by SEMAT to produce specific courses aimed at introducing the new standard [3] to upcoming software engineers. Software tools are being developed that leverage the standard's notation [4, 5, 6], which may readily be used at various phases of the training process of CS students. As a consequence of the methodical efforts in 2019, a groundbreaking publication emerges [7], which, in our perspective, is poised to swiftly establish itself as a definitive textbook for all university students pursuing software engineering studies. This study exemplifies the utilisation of Essence in implementing the field of software engineering for undergraduate students at the Computer Science Department of National Research Tomsk State University.

Soft Skills and Software Engineering

In the software engineering domain, various soft skills, such as collaboration, efficient use of time and resources, and risk mitigation, are intricately connected to complex and highly specialised aspects like software development processes, analysis and requirements gathering, software architecture design, software prototyping, version control, and team management, all while adhering to these processes. In addition, issues pertaining to the financial efficacy of project work, legal matters, and other factors further contribute to the collaboration with the client. Hence, in addition to the imperative of developing each of the aforementioned talents, it is crucial for prospective graduates to comprehend the interplay between various software engineering processes and their correlation with the overall project and team. However, it is exceedingly challenging to attain these talents through the educational process that relies on lectures and repetitious exercises.

To address this problem, we employed the project method [8] and completed the following primary tasks: firstly, we assisted students in gaining an understanding of contemporary methodological concerns in the realm of software development processes; secondly, we demonstrated how theoretical studies manifest in actual development processes; thirdly, we afforded students the chance to engage in a software development project; and finally, we formalised this experience by utilising a project management environment. Nevertheless, despite resolving these issues, the challenge of reconciling divergent approaches to managing software development projects persisted within the development process sector. Consequently, as a consequence of this issue in prior years of teaching this subject, teachers had to restrict the students' options by only providing the Unified Process, which encompasses the majority of corporate software development processes. Nevertheless, the relatively recent introduction of SEMAT and Essence has enabled us to enhance this field, rendering it more adaptable without compromising its core principles.

Overview of the Research Topic

Within the framework of undergraduate education, the subject in question encompasses a sequence of courses focused on object-oriented software development technologies. Therefore, students who possess knowledge and practical abilities in areas such as object-oriented programming, object-oriented analysis and design, and software system architecture are the ones that study this topic.

Prior to commencing this course, the instructors must select a project and a client for the duration of the semester. The customer chosen for this project is a non-specialist in computer science, deliberately selected to prevent any bias in providing suggestions for improving the system. Additionally, the customer has expressed a desire for a similar product to assess the practicality of the students' developed system. The project's level of complexity should be about equivalent to the effort of two individuals working full-time for one month. In order to execute the project, students are organised into teams consisting of 6-10 individuals. The specific number of team members is determined by the total number of students enrolled in the course and the customer's willingness to collaborate with a certain number of teams. Every team possesses an own project manager. Generally, a project entails either a single intricate risk (technological or analytical) or many moderate-level challenges. Teams are required to provide updates on the project's advancement through the utilisation of the Redmine project management platform, which has been customised for educational use. During the project, students engage in a simulated real-world IT company experience. They undertake specific tasks, assume designated roles in the development process, interact with other participants, and accumulate points (representing virtual currency) that are subsequently converted into grades.

Lectures

The purpose of lectures shifts as the primary objective of the course focuses on project implementation. These types may be categorised into two distinct groups: factual classes, which focus on theoretical aspects of contemporary software development, and practical classes, which include analysing the mistakes made by teams inside the project framework. The use of the methodological foundation of Essence enables us to examine many practices and techniques of software engineering using a unified descriptive language in the lectures of the first category. This enables students to have a comprehensive understanding of different software engineering processes and establish a theoretical foundation for evaluating and contrasting different approaches. Teachers offer practical lessons, which are centred around a common project that all teams work on. Consequently, this approach provides more clarity for all students, as all teams engaged in the same project have comparable challenges. The visual components of the Essence language, which we have largely incorporated in the project management system, are the most effective way to convey problematic circumstances.

The key component of the implemented Essence language is the Alphas desk, which represents the abstract degree of project health for a single project. This desk facilitates the demonstration of several project stages and provides guidance to student teams on how to accomplish the key milestones for the course. Teachers may organise all 7 standard Alphas and their corresponding states in separate columns on this desk and manipulate them using checkboxes.

Practical Classes

During the practical courses, the lecturers assess the present condition of their project with each team separately. Teachers do not actively involve themselves in the customer's system development process, meaning they do not assess the code, create the architecture, or write any code. The students are mostly responsible for this portion of the task, but professors may offer assistance in the event of a team crisis. However, such occurrences are few, happening on average once every semester and only affecting one team. The lecturers primarily emphasise the students' ability to structure their work and effectively apply the knowledge acquired to development processes and project management. Practical classes are separated into two successive phases: an introduction phase and a work phase.

The teachers' responsibility during the initial phase is to demonstrate potential outcomes of selecting specific methodologies for the team. pick instance, when a team opts pick one of the Scrum versions for their work, it becomes their responsibility to document client needs using the Product backlog and modify plans on a weekly basis (experience has shown that managing weekly iterations is the most convenient approach). In this format, the teachers are responsible for elucidating the distinction between the outward features of a practice and its substance. They must demonstrate to students the disparity between the generalisations that depict the procedures and the actual manifestation of the practice throughout the project. Each team undergoes the initial phase in a unique manner, with some teams taking up to 2 weeks to independently identify and address the majority of the key challenges. One team required a duration of one month to do this task, highlighting that the project had already commenced and the deadline for its completion was imminent. In Essence language, the primary objective of professors at the beginning of student projects is to assist students in selecting and establishing their working methodology.

Once the team rectifies its Way of Working, the work phase commences, when the practical classes transition into consultations and expedited assessments of the project's progress. Depending on the approach chosen by the students, various project challenges may be handled by determining the most effective methods for addressing them.

CONCLUSIONS

Previously, we said that the primary impact of utilising Essence to alter this topic was to enable students to select diverse practices and methodologies of software engineering, while also enabling the instructor to present a comprehensive overview of other aspects of development. However, as Essence is still evolving, we would want to highlight other outcomes that arise from utilising Essence as an instructional tool for students. Overall, the use of Alphas has proven to enhance students' abilities and comprehension, as evidenced by our findings from conducting personal interviews with students. However, there appears to be a challenge associated with the Essence work items that are linked to the matching Alphas.

1) The Specifications and Software System The pupils have a thorough comprehension of the work products. The majority of software engineering approaches and practices primarily concentrate on the solution domain. The students can readily comprehend the process and gauge their advancement at each phase of the project by utilising well recognised and comprehensible Work Products templates and examples.

2) The Opportunity and Stakeholders Alphas pose a greater challenge for students' comprehension. To comprehend this level, it is necessary to attend an extra lesson on corporate objectives and broader stakeholders, beyond merely the users. However, there are additional challenges associated with the Work Products. Students can access beneficial Work Product templates for Stakeholders, such as contacts, user profiles, and UX user profiles. However, they encounter difficulty in locating high-quality Work Product templates for the Opportunity Alpha. These templates would assist in formalising project information and enable the monitoring of progress in the Alpha state.

3) The pupils find the Endeavour area of concern particularly challenging to comprehend and use. The Alphas are commonly seen as indicators of the team's or company's overall success, rather than specifically reflecting the development of the project. Unlike previous Alphas, the student teams specifically struggle with the Work Products examples in the Endeavour area of concern. Alternatively, when you do an action, such as creating or updating a Work Product, it might lead to advancing the Alpha states. However, teams in the Endeavour area of concern have challenges when it comes to creating Work Products. Due to the absence of widely recognized and universally recognized templates for the Alphas, particularly the Way of Working, there is a lack of established Work Product frameworks. In order to meet the requirements of the topic, the teacher requires students to create a work product known as "Team rules" on the Redmine wiki. However, students are facing difficulty in finding a suitable example of a work product with a similar objective for actual firms. However, in essence, it follows a recursive principle: in order to progress the project's state in the Way of Work domain, it is necessary to compose a Work Product that outlines your team's Way of Work. This process can be perplexing at times and can result in difficulties for students in comprehending this particular domain.

Furthermore, users have the option to download plugins (https://www.redmine.org/plugins/semat_alpha_state_cards) that we have built to include Essence features into Redmine. These plugins may be utilised for educational or business purposes.

REFERENCES

Jacobson I., Meyer B., Soley R. "Software engineering method and theory — a vision statement", Feb 2010

SEMAT. 2013. Essence – Kernel and Language for Software Engineering Methods. Available at: https://www.omg.org/spec/Essence/1.2/PDF [accessed February 5 2019]

Zapata C., Jacobson I. "A first course in software egineering methods and theory". Dyna rev.fac.nac.minas [online]. 2014, vol.81, n.183, pp.231-241. ISSN http://dx.doi.org/10.15446/dyna.v81n183.42293.

Essence Enterprise 365. [Online]. https://www.ivarjacobson.com/essence-enterprise.

SEMAT Essence Kernel Tool https://semat.herokuapp.com/

Graziotin, D and Abrahamsson, P 2013. A Web-based modeling tool for the SEMAT

Jacobson I., Lawson H., Ng P.,McHahon P., Goedicke M., "The Essentials of Modern Software Egineering", 2019

Project method Encyclopedia of Educational Theory and Philosophy, ed. D.C. Phillips. Thousend Oaks, CA: Sage 2014. Vol. 2. Pp. 665-669.