THE IMPORTANCE OF RESTRUCTURING SOFTWARE ENGINEERING EDUCATION STRATEGIES IN ORDER TO MINIMIZE THE GAP BETWEEN ACADEMIC SUPPLY AND INDUSTRY DEMAND IN SOFTWARE ENGINEERING FIELD

MOHAMMAD SHKOUKANI, RAWAN ABU LAIL

1 Asstt Prof., Department of Computer Information Systems, Applied Science University, Jordan
2 Asstt. Prof., Department of Computer Information Systems, Philadelphia University, Jordan
E-mail: 1 m.shkokani@asu.edu.jo, 2 r_abulail@philadelphia.edu.jo

ABSTRACT

The field of software engineering is changing rapidly, this paper aims to present the history of software engineering education over the years and the changing happened to software engineering field. It focuses on the new methodologies used to develop the modern system that need special requirements and skills mainly in Agent-oriented software engineering and Component based software engineering. The paper highlights the importance of multi agent systems in building modern application, and it presents the gap between academic supply and industry demand in Agent-oriented software engineering. Finally it recommends strategies to restructure software engineering education to minimize that gap.

Keywords: Agent Oriented Software Engineering, Software Engineering Education

1. INTRODUCTION

Software applications and systems are well known and necessary in everyday life; there is a need for well-educated and skillful graduates in software engineering, who have the ability to build new systems and applications by applying new software engineering approaches. These new requirements need a review for software engineering education in order to make the software engineering education outcomes compatible with the new requirements for the new software applications and systems [15].

Because of that it is important to briefly look at how software engineering education has evolved over the years, and to mention the organizations that play roles in that evolutions mainly in curriculum design in software engineering in order to fulfill the needs of a rapidly developing field [9]. One of the most important factors that influence the evolution of software engineering education is the changing that happened to nature of the new applications of software engineering which required new features, specific skills and new software engineering methodologies and strategies [8], [12]. These new requirements and strategies were not supported by the classical way of teaching software engineering. So there must be new approaches in teaching software engineering in order to fulfill those new issues [8], [16].

While classical software engineering education focuses primarily on technical skills, the actual needed skills in the new systems and applications are different like team skills, communication skills and problem solving skills. The new trend in software engineering applications expects the systems to be autonomous, extensible, flexible, robust, reliable and capable of being remotely monitored and controlled [15].

These new features of systems require new approaches for building software engineering applications [20], [27]. Component-Based Software Engineering (CBSE) approach has emerged as a new software engineering approach for building these applications then there were refinements happened to that approach to produce another one, which was Agent Oriented Software Engineering (AOSE) [4], [5], [6], [7]. AOSE has emerged as a viable approach facing the new features needed in software engineering application and also it is now a cornerstone of enterprise software engineering [23], [26]. AOSE extends Component-Based Software Engineering (CBSE), and increase the flexibility, adaptability and autonomy of systems on one hand and on the other hand it reduces the
development cost of systems and also the required time for development. Also it improves the reliability, maintainability and overall quality of software systems [5], [7].

There is a lack of awareness for these new approaches in software engineering in education, which leads to weaknesses in the quality of software engineering graduates [15]. Because of that gap between the education in software engineering and the requirements in software engineering industry there is a necessity to restructure the education strategies in software engineering to narrow that gap.

This research is structured as follow, firstly it introduces the software engineering history from seventies until now, and then it presents the actual approaches in software engineering which are used in building the modern systems like AOSE and highlighting the differences between that approaches and classical approaches in software engineering. After that it focuses on the gap between academic supply and industry demand in software engineering field and suggests a solution to narrow that gap by restructuring software engineering education strategies, finally it presents a summary for software engineering educators and graduates.

2. HISTORY OF SOFTWARE ENGINEERING EDUCATION

First of all the researcher will introduce the evolution that happened for software engineering education over the years. There were three main periods of software engineering education history [9]. The first period was before 1978, the courses which were taught at universities in that time were free standing courses in software engineering. From 1978 to 1988 which represents the second period, there was introduction of graduate courses. Then the number of graduate courses was increased rapidly in the third period after 1988. Software Engineering Institute (SEI) was one of the organizations that played a role in the evolution that happened for graduate courses [10].

The design of curriculum for graduate course were also changed, they were firstly including areas like management issues. Then the trend of ideas was making the projects compatible with the rapidly developing field. But there were no serious efforts to implement these ideas. There was a workshop in 1988 by SEI Curriculum Design; the most important achievement of that workshop in software engineering curriculum design was the draft of first graduate course [11].

The committee suggested six core subject areas namely software systems engineering, software project management, specifications of software systems, software system and validation, principles and applications of software design and software generation and maintenance. Committee recommended that project work devoted by 30% of the degree program. Committee suggested also that electives may be included from system engineering subjects, computer science and software engineering [11].

Many universities after 1988 offered graduate programs. ACM and IEEE are professional organizations played a pivotal role in the evolution happened for software engineering education. These organizations have concerned about the needs of education and research in software engineering by conducting conferences, workshops and publications [24]. Another significant achievement of ACM/IEEE was Software Engineering Coordination Committee (SWECC) establishment, it was established to acts “permanent entity to foster the evolution of engineering as a professional computing discipline” [10].

Software Engineering Body of Knowledge (SWEBOK) was one of the most important projects of (SWECC). This project aimed to define boundary of software engineering as a discipline, and to provide the foundation for curriculum, licensing and certification. [9], [11], [16], [17], [18], [19].

Conducting Software engineering education is very difficult because the nature of software engineering field which changing rapidly. Software engineering students are required to be skillful in programming, computer architecture, system and application design. Graduates of software engineering is faced with massive changes in software engineering industry and software world, and faced with different required skills. Aside from technical skills, software engineering graduates are also required to master another type of skills like social skills and soft skills, which enable the graduates to work with teams and improving the way of presenting knowledge.

Classical software engineering education curriculum focuses primarily on technical skills, the new approach of teaching software engineering must include the required soft skills and increase the practical skills of students by placing them in some situations similar to the situations they are supposed to face in the professional lives [8], [12], [15], [16].
3. AGENT ORIENTED SOFTWARE ENGINEERING EDUCATION

Nowadays there is a need for a wide variety of enterprise and mission-critical applications like communications, computing, air traffic control systems, space, defense, manufacturing, health care, highway safety and e-commerce. These modern systems are different from classical systems because they required to be extensible, robust, reliable, autonomous, flexible and could be controlled and monitored remotely [20], [21], [22], [27].

These new requirements impose a key challenge for developing the modern systems. A new viable approach has emerged to address this challenge which is Agent- Oriented Software Engineering (AOSE) [1], [2]. This new approach is considered as a cornerstone for building modern applications [23], [26]. Because the necessity of this new approach it is important to involve this approach in curriculum of software engineering to make the student familiar with it. This can happen by restructuring software engineering education to make the outcome of universities compatible with the actual needs of real world [7].

Need of extensive selection of modern applications in software engineering like communications, computing, space motivated, the search for new development strategies and new methodologies in software engineering [26]. These new strategies and methodologies must support the development of modern applications that have the requirements of being reliable, autonomous, robust, extensible and able to be controlled and monitored remotely. Classical software engineering methodologies fail to achieve these requirements. The attractive alternatives for classical software engineering methodologies are new approaches like Agent-Oriented Software Engineering (AOSE) and Component-Based Software Engineering (CBSE). These new approaches are suitable for developing modern applications [24], [25], leading to greater adaptability, autonomy and flexibility. (AOSE) extends (CBSE), while (CBSE) has great potential in reducing the cost and time of application development, (AOSE) have much more features like improving maintainability, reliability, and the quality of software systems [4], [5], [6], [7].

We can say that the new generation of software components is software agent components (agents), which extended the traditional software components with greater autonomy, adaptability and flexibility. The methods and technology for agent-oriented software engineering provide a basis for engineering autonomic, self-managing, self-healing and self-tuning systems that are able to dynamically adjust and reconfigure themselves in response to changes in their environment [1].

Due to the different nature of AOSE, it needs new roles of software engineering and new competences which are significantly different from those in classical software engineering.

The new roles of software engineering within AOSE are classified under two namely domains as follow [4], [7]:
• Agent Component System Engineering (ACSE).
• Multi-Agent System Engineering (MASE).

ACSE concerns with the agent components development, while MASE concerns with the multi-agent systems development [4]. Different set of competences required in each domain. The lifecycle of ACSE is divided into five phases as follow: requirements analysis and determination of agent component, design of agent component, implementation of agent component, testing of agent components and finally maintenance and evolution of agent component. It is important to mention that the agent category determines the design for the agent component.

The mobile agents for example require concerns about security issues this leads to additional complexity in the design phase to deal with agent mobility. While the lifecycle of MASE is divided into six phases as follow: requirements analysis and determination of the multi-agent system, selection of a set of agents and customization, selection an architecture for multi-agent system and customization, integration of multi-agent system, testing of multi-agent system and finally maintenance and evolution of multi-agent system [6].

4. RESTRUCTURING SOFTWARE ENGINEERING EDUCATION

Due to differences between the software engineering skills, competences and roles, needed in agent-oriented software engineering, and those needed in the traditional software engineering, it is important to restructure software engineering education in order to meet the needed requirement in AOSE. Restructuring process will concentrate on life-long learning by building a solid foundation for it, and on software engineering curricula which must cover all new subjects in software engineering field; this could be happened in AOSE if there is integration between AOSE research and education [3]. This integration will narrow or close the gap between what software engineering programs offer.
and what are the actual requirements needed from graduates of the programs of software engineering. Also this integration or partnership between software engineering industry and software engineering educational institutions will resolve the problem of the lack of adequate preparation of software engineering graduates [16], [17].

The objective of the partnership and cooperation between industry and educational institutions is to achieve the goals for both parties and to minimize the gap between academic supply and industry demand [11], [12], [22]. It is important to teach students how to be productive in software engineering field, and learn them how to interact with the dynamic environment of software engineering, this can be done by academic institutions by making students working at teams by assignments and team-based projects [13].

Software engineering education must be flexible and responsive in order to face technological changes in software engineering field. Also there is a problem in the variance of speed in changing software engineering curricula and the changing happening in software engineering field. Changing speed in software engineering field is faster than changing happening to software engineering curricula at academic institutions [14], [28], [29], [30].

Software engineering education must be changed according to the changes in software engineering field and according to the new models in software development The programs of software engineering education are expected to produce an adequate software engineers for software engineering field who are well prepared and skillful to develop, deploy, maintain and evolve a wide variety of agent based systems, multi-agent systems mission-critical software systems. In order to produce an adequate software engineers there is a need for restructuring software engineering education. There are many strategies to restructure software engineering education, like integrating research and education, building a solid foundation for life-long learning and education, establishing industry-academic partnership in research and education [13].

5. CONCLUSION

The field of Software engineering is developing rapidly; this paper presented the history of software engineering education and focused on the difference between the outcome of universities and what the actual needs of software engineering field and how to make these outcomes compatible with the market needs.

It also focused on the curricula and courses that are taught at universities, and found that these courses are focusing on classical software engineering approaches, and there is a lack of awareness of the new approaches of software engineering.

Then it presented the importance of AOSE and CBSE in software engineering in building the modern systems and applications. AOSE and CBSE are new approaches in software engineering which have many features that are suitable for building new applications and systems in software engineering. The paper also showed that there was no attention from universities in these new approaches, and there is no courses supporting these approaches.

As the absence of this awareness at universities, it leads to weaknesses of the graduate's quality. This means that there is a lack of skills and information needed for those modern approaches, and thus it makes the graduates facing difficulty in dealing with the modern systems in the labor market. Depending on the above, the researcher developed a set of recommendations one of them was the importance of restructuring methods in software engineering education so that there is continuous development of educational plans so as to be proportionate to the requirements of the labor market.

6. ACKNOWLEDGMENTS

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the partial financial support granted to cover the publication fee of this research article.

REFERENCES:


[10] www.swebok.org / date of access 11-1-2012


