Vol. 6(19), Jan. 2016, PP. 2709-2718

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

Developing a Teaching Framework to support software inspection

Nasim Gazerani [1*], Navid Gazerani [2], Rodina Ahmad [3] and Soroush Forouzani [4]

[1,3,4] *Department of software engineering, University of Malaya, Kuala Lumpur, Malaysia*

[2] *Department of computing, UCSI University, Kuala Lumpur, Malaysia*

***Corresponding Author's E-mail:** rodina@um.edu.my

Abstract

The objective of inspection process is to reduce the cost by finding and removing defects earlier. In recent years, there have been a number of attempts to further increase inspection efficiency by the introduction of tool support and resulting in a number of prototype systems. However, many software engineers suffer from lack of background knowledge and experience on software inspections and its techniques. The purpose of the study is to investigate the possible ways to teach software inspection processes in both sides of concepts and practice to Software engineering students. So, we offered a teaching framework to make software engineering students have a deep understanding on software inspection and to improve their practical abilities. The framework consists of three parts: general guidelines, specific guidelines and learning activity. In addition, there are two parts of general guidelines: Software inspection concept and Software inspection technique. Specific guidelines include Software inspection process and applying technique on Software inspection process. The third part includes conceptual samples, collaboration in practice, work sample and assessment for improving practical skills and abilities. Based on the teaching framework, a software support tool is designed and developed using interactive feature such as combining text, sound, graphic, images and animation. It is evaluated by software engineering students in University of Malaya to show positive impact of framework on teaching software inspections.

Keywords: *software inspection; guidelines and teaching framework*

1. Introduction

Performing software inspection is the industry's best practice to deliver high quality product. Software inspection is an essential stage for managing software defects early in the development process (Sapsomboon,1997). Software defects are some of the most important factors that affect the success of a software project. For over 20 years, software inspections have been described as one of most effective quality assurance techniques in software engineering (Laitenberger & DeBaud, 2000). The initial goal of an inspection is to identify defects early as defect heavily affects the improvement of the overall quality of software, with consequences that involve costs and time (Yourdon, 1997). Inspection is a process that it consists of checking by a group of people with specific roles. The inspection process usually assumes that each participant has knowledge and experience to inspect documents. Organizations prefer to employ experts to perform the inspection processes in order to gain better outcomes. The lack of sufficient knowledge of the software inspection among software engineers has negative effects on inspection process in organizations. Therefore, organizations would like to choose inspectors who are qualified enough to the related domains to be members of their Software

Article History:
Received Date: Aug. 12, 2015
Accepted Date: Jan. 16, 2016
Available Online: Feb. 06, 2016

Requirements inspection team. Software engineering students should be critical thinkers and should acquire the knowledge, skills and practical abilities that are related to software engineering subjects such as software inspection and its techniques. Most of the students aim to prepare themselves for their future career. The effective performance of the software inspection process requires some background knowledge as well as some experience in requirement engineering. Students also need to be able to work in a group on software inspection projects and learn how to perform the process, not only rely on context-based knowledge. Unfortunately, most software engineering students are not exposed to requirement inspection process practically in their studies.

As a result, the software engineers should be taken by surprise when they work in organizations as inspectors because they do not fulfill the organization's expectation and also cannot solve the problems. On top of that, it is time consuming for organization to ensure that software engineers are familiar with software inspection and its technique. Training them would raise the total cost of software development. In this study, we propose a teaching framework that aims to support software engineering students to have a deep understanding of software inspection concepts, process and to improve their practical abilities. We also developed a teaching tool based on the proposed framework, which is called Web-based Teaching Software Inspection Tool (WBTSIS).

The teaching framework guides software engineering students to overcome the complexity of software inspection concepts, to obtain appropriate learning skills and to improve practical abilities. It focuses on three measures general guidelines of software inspection, specific guidelines and learning activities based on blended learning theory.

The WBTSIS is evaluated by 15 software engineering students to show its usefulness and effectiveness. The paper is structured as follows: in section2, literature review is summarized and discus. The section3, rework is presented. Section4 present the WBTSIS tool and shows its use according to underlying inspection process and its techniques. Section5 describes experiments where the WBTSIS tool was evaluated by software engineering students. Finally, section 6, we provide conclusion.

2. LITERATURE REVIEW

A. Software Inspection

A.1. Introduction to Software Inspection

The main concept of inspections is to find faults in the software document. Inspection helps to ensure that the artifact has sufficient quality. The main objective of inspection was defined by Fagan. For over 20 years, software inspection has been proposed as a simple, cost-effective technique for defect detection in all types of documents such as requirement, design, code and test. The primary aim of an inspection is to find faults early and then enhance the general quality of software within the desired time and budget to achieve best results (Laitenberger and DeBaud 2000). The inspection introduces an approach that includes a well-defined process and it can analyze a software product using a specific technique such as reading technique to satisfy the objective of identified faults. It means that the author can apply inspection on a requirements document, design, or code.

A.2. Software inspection Process

Software inspection contains six important steps: planning, overview, defect detection, defect collection, rework and follow up. However, these steps are common for the inspection process (Laitenberger & DeBaud, 2000). The first step is *planning* that it consists of defining objectives, identifying inspection participants, their assigning roles and making a schedule for inspection meetings. (Laitenberger & DeBaud, 2000). During *overview*, a moderator makes inspections easier and to make it

clear for participants to understand. (Laitenberger & DeBaud, 2000). During defect detection, each inspector analyses the document to become familiar with it and individually find possible defects. Fagan (1976) states that a group meeting gives positive results, that is, most defects are found because inspection group members discuss and analyze the document together. During defect collection**,** team members participate in the inspection meeting to examine a software document in order to identify defects. Hence, each inspection participant should collect and record the defects detected. Then, a decision must be made on whether a defect is really a defect. (Laitenberger &DeBaud, 2000). During rework**,** the author revises and fixes defects detected by the inspectors (Fagan, 1976). He or she makes corrections to the inspected document and investigates each reported defect. During follow up, the moderator investigates whether the author has revised all found faults. The moderator then gives final suggestions for the collecting process and the product data for the proposed quality improvement.

A.3. The Reading Technique

The most vital and difficult part of requirement inspection is finding defects. In this part, the inspectors find faults. To perform this step, the inspector can apply a reading technique for supporting inspection (Aurum & Wohlin, 2005). A reading technique demonstrates a sequence of stages that guides an inspector in gaining a deep knowledge of the requirements related to inspection and identifying issues related to them (Laitenberger, 2001). The inspector can meticulously identify defects carefully by applying particular techniques in software documents that may enhance the efficiency of an inspection team.

B. Blended Learning

As is known to all, teachers and students are the major factors in teaching and studying process. In the traditional mode, teachers usually plays an important roles and students act as subordinate subjects. Either what to learn or how to learn are all decided by teachers. The students have to accept passively. In recent years, the integration of multimedia technology, network technology in global education domain has promoted the Constructivist Teaching Theory prevail in many countries.(HOU Shuang, 2008) Using this traditional teaching mode, students' abilities are often limited into a narrow range. The traditional approach to education, where the transfer of knowledge is achieved mostly by lecturing, has a number of shortcomings, in particular because the students are not motivated enough to acquire knowledge actively. The Constructivist Teaching Theory is not adaptive to current education conditions as well. Students often feel inconvience because they cannot gain enough guidance from their teachers using this way. In the view of this, the proper method is to combine both of them, which can not only enhance teachers' leading role, but also highlight the prime positions of students. In another word, it is Blending Learning.

Blending Learning", means combining the advantages of traditional learning mode and E-learning mode synthetically, giving teachers play on guidance, inspiration and control, at the same time encouraging students to show initiative, positivity and innovation. The goal of blended learning is to join the best features of in-class teaching with the best features of online learning to promote active independent learning and reduce class seat time. Utilising computer-based technologies, instructors can use the blended model to redesign some lecture content into new online learning activities, such as tutorials, selftesting exercises, online group discussions and collaborations. Blended learning offers a number of advantages over traditional learning (i.e. face-toface teaching) and totally online learning. Blended learning allows the instructors to accomplish course learning objectives more successfully than traditional courses do. It also gives instructors more flexibility with their classes. It increases interaction and contact among their students and between the students and themselves. Students will be more engaged in learning activities and therefore will seek out more assistance.

3. RELATED WORK

This part begins by describing the existing teaching framework to support software inspection. There are three main types of teaching frameworks that help students learning.

A. Teaching framework of ERP courses

Deng designed a teaching framework for ERP courses that helps students gain a deep understanding of ERP and to improve their practical abilities. The framework consists of three parts: experiments in the lab, extracurricular activities and internship in a certain software company. Furthermore, are three levels of experiments in the lab: business scenario simulation, business process and roles collaboration designed according to Zachman's enterprise architecture framework. After the experiments in the lab, students will be ready for training and can obtain basic knowledge of it. Extracurricular activities include the research of ERP business game, the simulation of development and testing of some ERP modules, and the implementation solution design of ERP. These activities will help students to improve ERP skills. Students' jobs during internship may be system development, system testing, system implementation and customer service that helps student gain achievement and enhance chances when hunting for the jobs. The framework has been applied in real teaching environments and has gotten good outcomes (Deng, Yin et al., 2009).

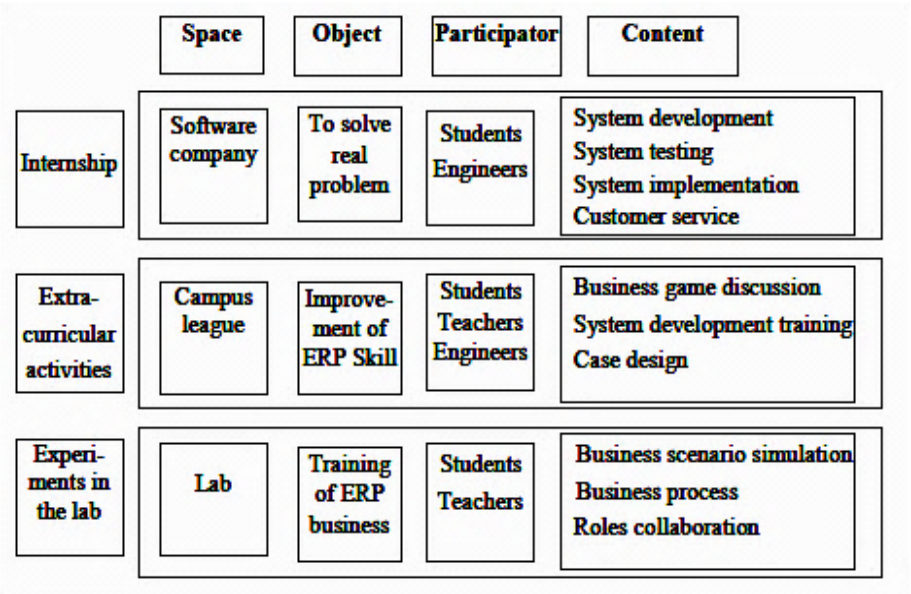

Figure1: The practical teaching framework of ERP course *(Deng, Yin et al., 2009*

B. Skill based framework for industry career

A common complaint by industry professionals is the lack of knowledge of Software Engineering (SE) graduates. The lack of skills of many new graduates directly effect on industry's progress. For example, some of the graduates do not possess communication and managerial skills. Internship in huge industry based heaven projects are motivation for undergraduates who are in the penultimate and final years of their degree courses as they prepare to embark on their professional careers. Although, the goal is to provide students with a real-life SE experience and to prepare them for industry, these projects failed in some cases to provide the necessary breadth of skills. This failure motivated the researcher to develop a teaching framework based on skills. It focuses on managerial, engineering and personal skills. This new experience shows that this framework guide student learning during projects. High scores in student

feedback questionnaires indicate that the skill-based framework helped to guide student learning outcomes through specifying clear objectives. Overall, the application of the framework was a success from the perspective of students, staff and clients and fulfilled its major purpose of providing students with an adequate breadth and depth of software engineering skills. (Karunasekera and Bedse, 2007).

C. A Teaching Framework for Essential Topics in Electromagnetics

Teaching undergraduate engineering electromagnetics (EM) requires extensive presentations of the basic theoretical and fundamental physical concepts that support most electrical engineering principles (Cheng et al, 2003). The following challenging components of teaching undergraduate EM courses were identified. Firstly, the technical level of the material makes it is difficult to teach students on how to use basic electromagnetic principles. Secondly, motivating students to learning EM poses many problems. Finally, in the process of learning the introductory essentials, it is not easy to link the concepts they are learning with applications and practice. Therefore, a teaching framework proposes to help overcome the mentioned obstacles and challenges encountered in instruction of introductory EM courses. The framework creates connections between theoretical fundamentals and pedagogical examples, and between historical perspectives and courses and the practical applications found in industry (Giannacopoulos and Popovic, 2005).

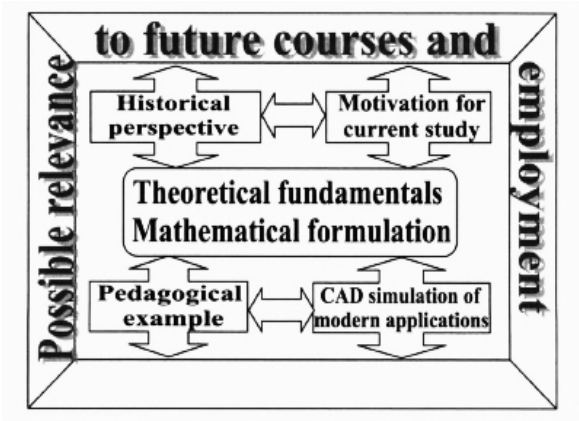

Figure2: Conceptual sketch of framework illustrated as a concept map

4. THE PROPOSED TEACHING FRAMEWORK

The framework consists of three parts: general guidelines, specific guidelines and learning activity. In addition, there are two parts in general guidelines namely, Software inspection concept and software inspection techniques to obtain general knowledge. Specific guidelines include Software inspection process and applying the techniques on software inspection process. The third layer includes conceptual samples, collaboration in practice and work samples for improving practical skills and abilities. The framework will be explained in detail with illustration in the following pages.

A. General Guidelines

The teaching tool is presented to support students learning about software inspection. It enhances the efficiency of software inspection learning that it is difficult to understand. This framework helps software engineering students to learn about when, where and why to use the knowledge they are learning. The section on general guidelines is designed with two components, software inspection concept and principle, and software inspection techniques, as shown in Figure2.

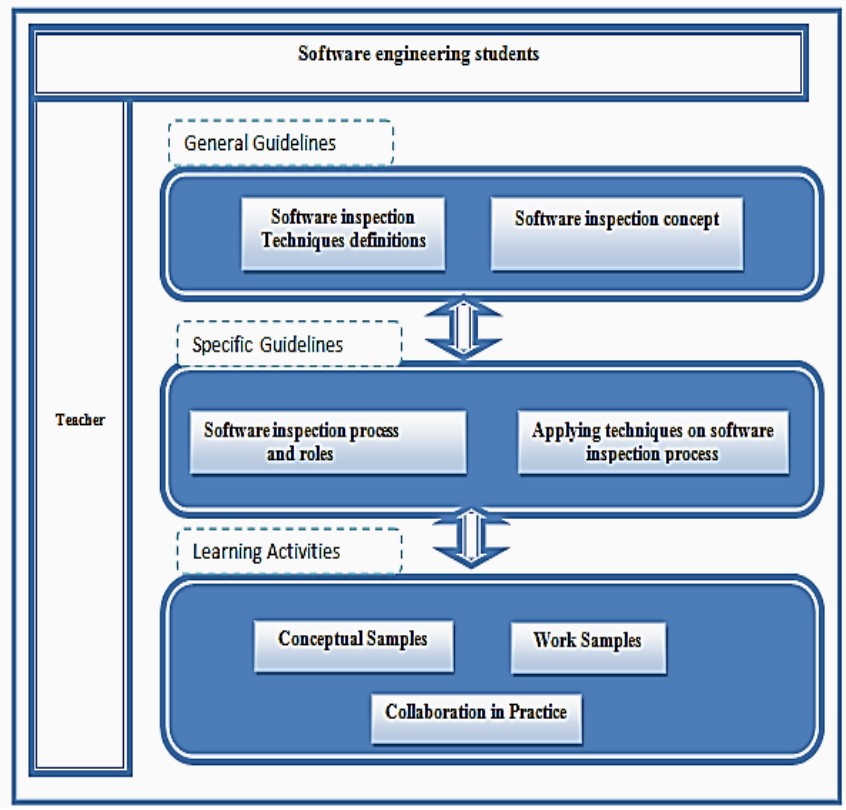

Figure 3: Teaching Framework for software inspection

A.1. Software Inspection Concept and Principle

In order to have effective way of teaching, ambiguous and difficult definitions must be avoided and a basic explanation of software inspection given. The concepts should be taught to software engineering students using a combination of multimedia method such as combining texts, sounds, graphics, images, animation and interactive features, as it is the best way to achieve efficiency and high quality of teaching. This teaching methods improves efficiency in teaching and motivation software engineering students to learn, be more interested in learning about software inspection, and be eager to answer questions such as why Software Inspection is important; when can software inspection be used; and how software inspection should be used. In this section, students will achieve primary knowledge of software inspection and its concepts.

A.2. Software Inspection Technique

It is imperative that students well understand software inspection technique because it is the key activity for defect detection. Such techniques are referred to as reading techniques. After learning about techniques, students will be more interested in thinking about software inspection and eager to apply the techniques for software inspection. The techniques will help students to identify defects in software documents which will increase the effectiveness of an inspection team (Laitenberger and DeBaud, 2000). The most popular reading techniques are ad-hoc reading and checklist-based reading.

B. Specific Guidelines

This section can be considered as the main part for students' understanding and consist of in two components, software inspection process and roles and applying technique on Software inspection

process. A more specific focus is given to this section on finding defects and selecting appropriate techniques so that students will be able to efficiently find defects and execute tasks using specified techniques.

B.1. Software Inspection Process and Roles

Using the general guidelines, students will become familiar with the primary Software inspection concept. In this part, the software inspection process is presented step by step using teaching tool help make understanding easier. This component of general guidelines teaches students about team roles and team size for the software inspection process. This component also indicates how many persons are assigned to each role and how to select people for each role. The next step involves each person doing his/her task in software inspection process based on his/her specified role, such as analyzing the document, identifying defects and correcting them. Some students may face many problems and have question about this part. To solve students' problems, this part should be taught step-by-step. To support learning by students who have different level of understanding and knowledge of software inspection, from the easy stages to the more difficult.

B.2. Applying technique on Software Inspection Process

Techniques are used for supporting defect detection in a requirement document and may increase the effectiveness of an inspection team. The teaching tool demonstrates how a technique can be chosen and how it can be applied on particular document based on this framework. The techniques help a student to scrutinize a document for identify defects. Finally, students can answer questions how techniques can be applied to software inspection and which technique is useful. This part is very critical and difficult to learn because identifying techniques is not easy and needs to more concentration, background information and skill.

C. Learning Activity

For software engineering student learn effectively, the learning activity has to be practical as it means that there are more opportunities to practice. There are a lot of samples to evaluate students' practical abilities about the software inspection process. The main goal of this part is to build a flexible model on software inspection and its dimensions by using a lot of conceptual and work samples, as well as collaboration with other students. This framework prepares a way for both experts and novices to improve their knowledge and practical abilities. Learning activities help students to improve their understating and abilities about concepts and practice, whether in groups or as individuals and students can also evaluate their obtained knowledge. Students can practice based on knowledge that they gained from all guidelines and concept approaches. This framework improves software engineering students' creativity and initiative for executing tasks. Finally, they will gain experience for their careers and research in the industry.

C.1. Conceptual Samples

Using conceptual samples is an effective way to present all the concepts of software inspection. Conceptual samples include examples, tutorials and quizzes. They show students how they can gain software inspection knowledge and where they can use it. The method prepares students for the practical step of learning as students will execute all tasks individually or in groups. Quizzes help to evaluate the level of students' understanding about software inspection concepts. It is a good method for a teacher to evaluate students. Sometimes, it encourages students to learn better and to focus more

a concept and principle. They can also improve their learning skills. Students' thinking should be clearly elicited by using quizzes and feedback must be provided.

C.2. Work Sample

Project-based learning samples or case studies are needed to teach student how to identify defects. Students should do inspection processes on the case study or project individually or in groups. Students should analyze document, identify defect, discus about defects and correct them by specified techniques. If they face problems, they can refer to the guidelines and get help. They can send the results to the teacher for correction and get feedback.

C.3.Collaboration in Practice

Collaborative learning helps students to learn more effectively. Many educators place a high premium on teaching strategies that go beyond more mastery of content and ideas. Students collaborate with each other in group to learn the dynamics of working with each other and to take role for identifying and correcting faults in the document samples. Each group consists of five or ten persons. The member of these small groups chooses their special roles based on their knowledge and experience in those roles. Every group should finish their work on time before another task start. If a group member faces a problem and is unable to solve it, she/he can seek help from another group. The students do their tasks and create solutions according to their roles and responsibilities and finish process collaboratively. Extending this area of concern to professionals leads to increased project-based and team-based activities that are grounded on sharing of tasks and responsibilities. The end goal is to lessen individual competition that has been well entrenched in almost all working environments in different canters of learning and professional endeavors.

5. RESULTS AND DISCUSION

The software inspection definition and exercise have been used based on proposed teaching framework by online teaching tool that was ease of use with average functionalities. The students used to study the software inspection traditionally with limited work experience. Based on proposed teaching framework, the students appeared to enjoy studying concepts and working on projects collaboratively. We believe that the structure and components of proposed teaching framework is well-designed to teach software inspection. In order to assess effectiveness of our framework and obtain student feedback regarding the knowledge, skills and practical abilities for future improvements, a simple questionnaire was given to 15 software engineering students from the faculty of Computer Science of University of Malaya.

The important questions were 1) Did the framework help illustrate the significant examples and quizzes? 2) Did this framework make good learning environment for collaborative work? The questions and the summarized responses are given in Figure4 and Figure5. The responses ("disagree", "not sure", "agree") are qualified in Table1. I should mention all samples and guidelines were the same for the students. The framework is evaluated by students with designed teaching tool. The experimental results indicate that students obtained the desired breadth of skills and knowledge, which was the ultimate goal of the framework.

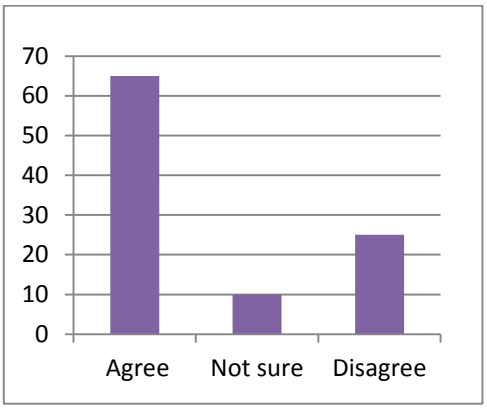

Figure 4: student responses to question 1

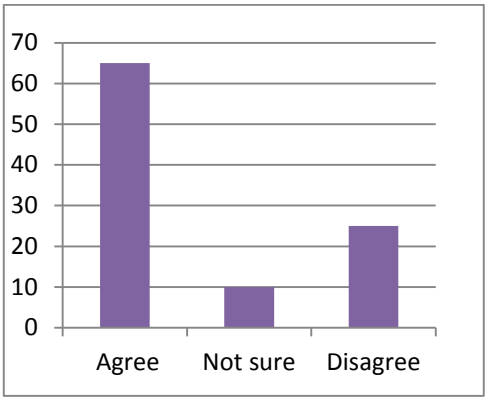

Figure 4: student responses to question 2

CONCLUSION

In this paper, we have presented teaching framework for software inspection. Studying on software inspection subject over the past years shows the lack of an appropriate teaching framework, because students don't have the necessary breadth of software inspection skills for industry. This motivates us to develop the teaching framework that aim to equip students with knowledge of software inspection to success in industry. This teaching framework was proposed based on blended learning that it explained. The aim of blended learning is to provide realistic practical opportunities for learners and teachers to make learning independent, useful, sustainable and ever growing. High scores in student feedback questionnaire indicate that this framework is helpful for student learning through specifying clear objective. This framework was a success from the perspective of students and fulfills its major purpose of providing students with depth understanding of software inspection skills.

REFERENCES

[1]. Aurum, A. and C. Wohlin (2005). "Engineering and managing software requirements." Technology and Management University of New South Wales Sydney

[2]. Dennis Giannacopoulos, Milica Popovic (2005). "A Teaching Framework for Essential Topics in Electromagnetics." Proceeding IEEE on Antennas and Propagation Society International Symposium,pp 117-120.

[3]. Karunasekera, S. and K. Bedse (2007). "Preparing Software Engineering Graduates for an Industry Career." Conference on Software Engineering Education & Training, IEEE.

[4]. Laitenberger, O. and J. DeBaud (2000). "An encompassing life cycle centric survey of software inspection." Journal of Systems and Software 50(1): 5-31.
[5]. Qingshan Deng,Aihua Yin, Baodong Tu (2009). “Design and Application of Practical Teaching Framework of ERP Course.” Proceeding IEEE on Computer Science & Education,pp1346-1349
[6]. Shuang Hou Shi-hong Chen (2008).” Research on applying the theory of Blending Learning on Access Database Programming Course teaching.” Proceeding IEEE on Education Technology and Computer, pp 396-398.