ENHANCING STUDENTS' SOFT SKILLS BY IMPLEMENTING CDIO-BASED INTEGRATION TEACHING MODE

Nhu-Hang Ha, Anand Nayyar

[1]Graduate School, Duy Tan University, Vietnam

Duc-Man Nguyen

International School, Duy Tan University, Vietnam

Chia-An Liu

EMBA Program, National Cheng Kung University, Taiwan

ABSTRACT

Currently, the software industries are paying crucial attention towards soft skills at the time of hiring new professionals, so there is a stringent need to enhance the role of the soft-skills in software development curriculum. During the four-year program, software engineering students are trained with a comprehensive amount of theoretical and applied coursework but soft skills training is not given significant importance. Moreover, students as compared to other subjects pay little attention towards soft skills. At Duy Tan University, with close vision to enlighten soft skills importance, we integrated two courses, which are focus on CDIO procedure (CDIO project level 1 (CMU-CS 297) and CDIO project level 2 (CMU-CS 397)), into the training program to offer students with practical experience in supervision, project management, quality control and decision-making. Student teams composed of 4 or 5 members are primarily responsible for solving game's problems (in CMU-CS 297) and developing a software project (in CMU-CS 397). In CMU-CS 397 course, the finished product is delivered upon project completion. The project evaluation is based on formal technical reviews of prototypes produced during the project life cycle basing on the stages of CDIO approach. Two instructors are appointed as mentors during course flow to support teams from conceiving to operating product. The objective of this paper is to present a field study in which 45 students are interviewed from these two courses to analyze their viewpoint regarding soft skills importance towards making themselves, successful professional software developers. The study is also conducted via group discussion with instructors to figure out the best possible information for research. The paper answers mainly two research questions: (1) What soft skills are appropriate for software developers, and (2) How instructors should possibly organize and conduct CDIO courses to enhance soft skills for students? After thorough analysis, that Leadership, Debate, Presentation, Teamwork, and Time management skills are the most valued for students. In CMU-CS 297 course, instructors should provide multiple types of games to enhance Debate and Time management skills of students. For CMU-CS 397,

mentors should encourage students to apply Scrum or Agile methodology to improve the Leadership skills for overall development.

KEYWORDS

CDIO-based integration teaching mode, Soft skills, Software Engineering, Teaching Methodology, Standard 3: Integrated Curriculum

INTRODUCTION

Dealing with human errors is much more difficult than technical problems for developers. The main cause of this trend is that human factors are usually related to soft skills such as teamwork, motivation, emotions, commitment, leadership, multiculturalism, interpersonal skills, etc. (Ahmed, Capretz, & Campbell, 2012). As a results, employers prefer to recruit developers who have both technical and nontechnical skills. While technical skills are relatively to evaluate by looking into academic credentials, certifications, professional experience, etc., which can be gained through education, training programs, certifications, and on-the-job training. Meanwhile, the non-technical skills focus on mentioned soft skills that are much harder to define and evaluate (Dumke & Richter, 2015). Recently, about 80% of the individuals who tend to fail at work because of their inability to relate or communicate well with others in a team (Kappelman, Jones, Johnson, McLean, & Boonme, 2016).

Software Engineering (SE) is a discipline deeply linking to practical aspects of developing software products within cost, schedule and quality requirements (Boehm & Sullivan, 2000). It requires software engineers to have technical and managerial expertise. Many learning and teaching methodologies are used in this field; however, most of the courses of SE are based on the classroom learning model (Ghavifekr, Rosdy, & Science, 2015). The teaching methods primarily focused on lectures and tutorials are not sufficient for SE students to develop the skills required for real-world problems solutions. Students must complete their tasks mostly on their own, in contrary to professional practice of team environment and collaboration. The need for modern approach to teaching SE is not new, but the supplying students with real problems and real teamwork environment is not really leveraged (Richardson & Delaney, 2009).

In addition, the curriculum of SE program is set in a real-world engineering context of a complete product lifecycle, i.e., conceiving, designing, implementing, and operating (CDIO), with design-build experiences integrated throughout the program (Vo, Nguyen, & Ha). The goals of CDIO include educating graduates with a deep and working knowledge of engineering fundamentals, who can lead in the development and operation of complex technical systems, and who have a strategic understanding of the role and impact of technology in society (E. Crawley, Malmqvist, Ostlund, & Brodeur, 2007). However, to enhance the soft skills of SE students we need to integrate other activities into the CDIO approach.

At Duy Tan University, we build up two courses focusing on introducing CDIO to freshmen and letting them getting familiar with CDIO spirit for further leveraging this method into advanced courses. The two courses are named CDIO Project level 1 (CMU-CS 297) and CDIO Project level 2 (CMU-CS 397). For CMU-CS 297, we provide various types of games for students to play in-group. During the time to solve the problem of the games, students will learn and upgrade their soft skills. For the upper level, in CMU-CS 397, instructors let groups

of students work on small software project. They are required to follow every step of CDIO approach. Thanks to the soft skills that are trained in CMU-CS 297, students are able to work well in their project with team members.

This study presents a case of the mentioned CDIO courses in the SE program at the International School of Duy Tan University. Data is collected from 45 students of the same course in the Fall semester of the academic year 2018-2019 by various methods such as on-site observation, historical report review, and structured interviews. The findings show the effectiveness of this integrated learning framework, especially in improving the soft skills of IT students. The experiences and lessons learnt from the implementation of this framework, along with challenges are also discussed in this paper.

THEORETICAL BACKGROUND AND CONCEPTUAL FRAMEWORK

Introduction to Conceive-Design-Implement-Operate approach (CDIO)

The CDIO is an open architecture endeavour that is specifically designed for, and offered to, all university engineering programs to customize and adapt to their specific needs. The vision of the CDIO is to provide students with an education that stresses engineering fundamentals set in the context of Conceiving - Designing - Implementing - Operating (CDIO) real-world systems and products (Gustafsson, Malmqvist, Newman, Stafström, & Wallin, 2002). It aims at developing a new model for engineering education (Gustafsson, Newman, Stafström, & Wallin, 2002).

- The Conceive stage includes defining the need and technology, considering the enterprise strategy and regulations, developing the concept, architecture and business case.
- The Design stage focuses on creating the design, that is the plans, drawings and algorithms that describe what will be implemented.
- The Implementation stage refers to the transformation of the design into a product, including manufacturing, coding, test and validation.
- The Operate stage uses the implemented product to deliver the intended value, including maintaining, evolving and retiring the system.

The proposed implementation framework

A good software engineer need grasp the software theory and basic knowledge, but also must have a deep understanding of the software industry and software project. They also require having a solid innovation design ability, communication skills, teamwork spirit, ability of lifelong learning etc. To work together effectively, the members in software development team need to have different non-technical skills such as teamwork, time management, operational management and overall team management skill. Therefore, we integrate two courses of CDIO into the curriculum as follows:

- In the course CDIO Project level 1 (CMU-CS 297), students are encouraged to play games to enhance soft skills through providing solutions for assigned problems such as team management and time management.
- In the course CDIO Project level 2 (CMU-CS 297), students also work in group to deliver a software product by putting the fundamental knowledge in software engineering into the procedure of four steps: Conceive, Design, Implement, and Operate.

After students finish those two courses they are able to apply advanced knowledge in SE field to complete the project of other advanced courses and Capstone project 1 and Capstone project 2. The proposed framework is described in Figure 1.

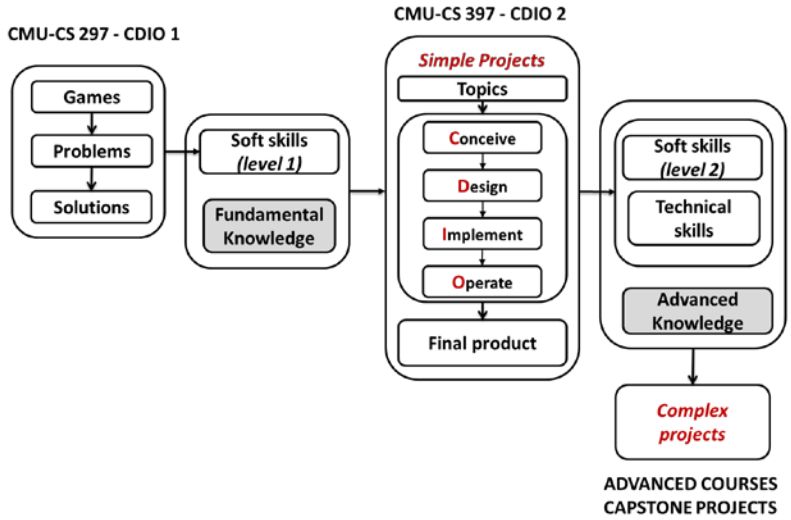

Figure 1. Proposed Implementation Framework

RESEARCH METHODOLOGY

The description of course CDIO project level 1 course (CMU-CS 297)

The course CMU-CS 297 consists of fifteen hours of classroom teaching in total, two or three hours per week for a duration of seven weeks in the second year. In this course, students are required to provide solution for the games. Instructors need to support them develop soft skills such as teamwork, problem solving, communication. Those skills are needed before entering the course CMU-CS 397. Teams are assigned topics to play an innovative game with pre-prepared materials. The games used for teaching and learning in this course are as follows:

Game 1: Think outside the box

Purpose: This game aims to let student consider different perspectives when solving problems. They can earn experience of problem solving in groups and add knowledge and information to each other.

Material: A copy of the brain strain hand-out (provided) for each player. The question of this game is “Without letting your pencil leave the paper, can you draw four straight lines through the following nine dots?”

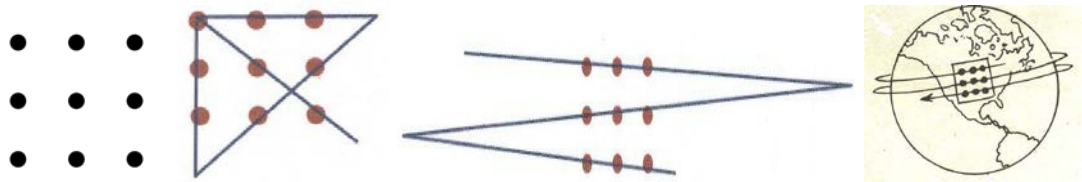

Figure 1. Think outside the box

Time required: 20 minutes

Procedure: Firstly, instructor deliver question with the photo to each player. Each individual has 10 minutes to look through the questions. Then instructor lets the players work in groups of 3 to 5 people to discuss to provide the solution.

Discussion questions: Before ending the game, instructor provides question to let students discuss:

- Working alone and in a team, what kind of performance do you find? What is the difference?

- What can you learn from your team members?
- How can we apply this to real life?

Reflections: The basic idea of this game is to build creativity, as the players need to challenge their own assumptions and look at things from a fresh angle. They need to break out of conventional thinking and take off the blinkers formed by experience. The usual way of presenting this problem is for a creativity trainer to give the first set of instructions. Once we start to think "outside the box", we open up many more possibilities and it becomes easy to solve the problem. In this game, team members must communicate in order to unify the topic, to assign appropriate tasks for each person, and debate with other groups to get the supporting or opposing. They can improve the skills of debating as well.

Game 2: Save the egg

Purpose: This activity is useful to illustrate the importance of teamwork. Ask everyone to reflect on how their group accomplished the task, what worked, what was challenging, etc. This team-building task gets teams working together, thinking creatively and managing their time.

Material: Raw eggs (one for each group plus extras in case of accidents), cardboard, duct tape, several thin straws (at least 40 per group), paper towels for clean-up, a way to enable a high drop.

Figure 2. Save the egg

Time required: 30 minutes

Procedure: The instructor divides the group into small teams of 3-4 students. Give each team one raw egg and other materials (depend on the instructor). Then he/she explains the rules of the teambuilding activity to tell them that the goal is to design and build a structure that will prevent their raw egg from breaking from a high drop. Teams will be given about 15 minutes to make the structure. If more than one team is successful, then the team that uses the least amount of materials wins. Before students start to work on their product, instructor gives 10 minutes to let them propose the ideas, design the product and explains the reason of their solutions.

Discussion questions: Before ending the game, instructor provides question to let students discuss:

- The designs changed or evolved over time.
- The traits or characteristics of good leadership or teamwork, or meaningful contributions during gameplay.
- Teams would do anything differently in the next time.

Reflections: Teams must work together to find a way to "save" the egg. That could involve finding the perfect soft landing, or creating a device that guides the egg safely to the ground. All members in team need to coordinate and assign tasks effectively to be able to complete

the product in the shortest time. After finishing the final product, every group has compulsory brief presentation to introduce their product designed to other groups for receiving the feedback-regarding positive/to be worked upon ideas.

Game 3: Build the paper tower

Purpose: The purpose of this game is to let student learn how to compete against other teams to see who can build the tallest tower. This is a great teambuilding activity involves creativity, coordination, and teamwork. The goal is to build the tallest or highest tower made of newspapers or paper.

Material: A measuring tape; for each team, also provide one stack of newspaper, 1 large roll, of masking tape, and scissors. Thus, an activity with four teams would require at least four stacks of newspapers and four rolls of tape.

Figure 3. Build the paper tower

Time required: 20-25 minutes.

Procedure: The instructor forms teams of 3-5 students. If necessary, the teams can be larger, but small teams are ideal to allow players to all stay involved. The instructor supplies each team with a stack of newspaper and a roll of masking tape. Each team will have a couple minutes to plan and discuss strategy, and then start a timer for 20 minutes. Each team will build a tall tower using the materials supplied. When time is up, instructor stop everyone and use the measuring tape to determine the winner. Towers must remain standing and not fall.

Discussion questions: Before ending the game, instructor provides question to let students discuss:

- What did you just do together? How did you feel while you did the activity?
- What was one positive thing that happened during this activity? What was one of the challenges of doing this activity?
- What did the group have to do or believe to be successful?

Reflections: Sum up the different ideas and feelings that students expressed, and restate ideas and learning moments the participants shared. Instructors provide examples of successful and unsuccessful design approaches to let students compare and learn the lessons. They need to have a clear vision and a plan of how to achieve it, then leadership skills can be developed.

The description of course CDIO project level 2 course (CMU-CS 397)

The course has a total of 45 hours, three to four hours a week for 12 weeks. To pass the course, a minimum of 80% attendance is required; in addition students have to submit a software product or a mobile application which is evaluated on the basis of CDIO stages

(Conceive-Design-Implement-Operate). In this course, students need to form their idea by themselves instead of getting form the instructor. Students can apply all skills, which were developed in the course CMU-CS 297 to fulfil the project such as problem solving, time management, leadership, and debate. CDIO is an outcome-based framework mostly for students in engineering and technology disciplines to develop real-world systems and products. This approach has three overall goals to educate students who are able to:

- master a deeper working knowledge of technical fundamentals.
- lead in the creation and operation of new products, processes, and systems.
- understand the importance and strategic impact of research and technological development on society.

In this course, students are required to manage a project using the learned principles in teams of 3-5 persons. The teams are self-selected by the students and every team has a team leader. The project teams can choose their individual project topics from a catalogue of ideas or choose other topics under supervision of the instructors. We have already set up an explicit criterion for this practice assessment based on stages of CDIO approach (Table 1).

Table 1. CDIO stages

Stage	Description	Required outcome
Conceive	Defining customer needs, considering technology, enterprise strategy and regulations, and developing conceptual, technical and business plans	Proposal, Requirements
Design	Creating detailed information, description of the design; the plans, drawings, and algorithms that describe the system to be implemented	Software/System Architecture, Database Design, User interface
Implement	Transforming the design into the product, process or system, including hardware manufacturing, software coding, testing and validation	Source code, Testing plan, Test case, Test report, Bug report
Operate	Using the implemented product, process or system to deliver the intended value, including maintaining, evolving, recycling and retiring the system	Demo Feedback Presentation

In the experiment guidebook, we make a list to let students know what they must complete and submit at each phase, and how they can get high scores. At each phase, we choose one team to present their process of project management in classroom; other teams can comment and ask questions for their presentation. At last, the instructor will review and summarize for the presentation. After the presentation, students should revise their project plan according to the comments and suggestions. At the final acceptance phase, every team should submit their project, documents and prepare for an oral defence. Students are assessed individually as well as in teams. Each team and every student must have an individual oral presentation for their work. A students' final course grade depends on their written reports, daily performance and oral presentations.

Students are more eager to produce something that has value beyond the classroom. Impressionable students are forming opinions of the utility of computer science and questions whether students would eventually change computer science study for another study with

better chance of giving them a career with some more noble purpose. By choosing a problem that is not within the experience of most students forces students to develop a high-level understanding and design before coding, as early implementation is not feasible. Through this experience, students get insight into the project requirements and constraints from the client perspective, learning how to overcome misunderstandings between clients and developers in terms of vocabulary, technology complexity and capabilities. Focusing on real-world projects in courses means that student's assignments do not have predefined problem or solution sets, which makes them harder to grade, but drive the students to extend their decision-making skills.

Data collection and analysis

For the purpose of this paper, we are using two different data sets. One set is the student survey. The other set is a focus group interview with 3 academic staffs who are the instructors of CMU-CS 297 and CMU-CS 397. We send out 100 survey and collect 45 responses. Most of them belongs to the International School, and 7% of them are in Faculty of Information Technology. The 80% of them are senior students and the rest of them are junior students. The ranking of the importance of the soft skills are presented in Figure 2. The highest score is 5.0.

- Teamwork skills (TEA)
- Presentation skills (PRE)
- Leadership skills (LEA)
- Time management skills (TIM)
- Debate skills (DEB)

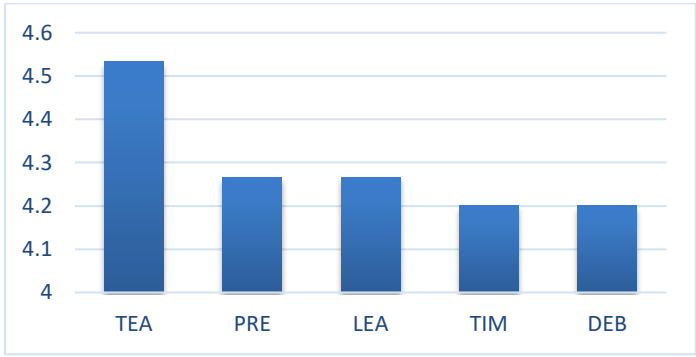

Figure 2. Ranking of soft skills

Students also provide suggestions to improve the CDIO courses. For CMU-CS 297, they want to have more games to practice the debate skill. Meanwhile, for CMU-CS 397, they need to get more knowledge on development software process and decrease overall CDIO's scope. To support them develop updated software, they suggests that instructors should add more lesson on new technologies or techniques and provide more free tools for software development.

After that, the authors conducted focus group interview with three instructors. The focus group interview was chosen since it would admit the instructors to interact as a group while describing their experiences of teaching at the program. In CMU-CS 297, there are some students do not really participate in the whole process of handling the problem (figure out the solution of the game or the project). In addition, students do not take much time for

proposing and designing idea. They start to solve problem upon receiving the question and correct the wrong solutions. Therefore, they do not have enough time to provide the best solution. The instructors need to change the evaluation scale and types. The peer evaluation should be added in to let other member grade themselves. The size of group should be reduced from five to three members to let them work closely and control the time. In addition, before working in team, students much to figure out solution individually to enhance their creative skills.

In CMU-CS 397, the instructor found that most projects are implemented at the basic level only. They lack of the tools to implement ideas such as programming languages, technologies. The team leaders still are unable to manage the members and assign the tasks effectively, and then they have not much time operate and test products. The instructors need to provide material in advance to let student read before coming to class. Students are also encouraged to spend more time on the project at home. Since within 4 hours at class they cannot fulfil all stages of CDIO approach. The instructors also suggest the management board provide more space for this course to let students have environment to work on their project together. Moreover, the English skills also become the obstacle for students to research the materials and work on modern technologies. The commitment of each group member is not high, which leads to the delay of the progress. Instructors suggest having teaching assistant to participate in technical guidance and assistance will increase the cohesion and curiosity of the juniors behind. It is necessary to organize CDIO product contests to award a monthly award to encourage creativity and scientific research of students.

DICCSUSSION AND CONCLUSION

To enhance soft skills of the students, not only for their professional development but also for real-world problems solutions and interaction is utmost necessary. To facilitate soft skills enhancement among the students, CDIO based course is introduced in CMU-CS 297 and CMU-CS 397. For CMU-CS 2917 course, multiple games are played for enhancing time management and soft skills and Scrum or Agile Methodology with real-world problems-based projects in team work are used in CMU-CS-397. This paper proposes modified framework for both courses by adopting CDIO based framework. And the study conducted on 45 students indicate that students have better experience towards soft skills, high team coordination towards problem solving and efficient utilization of technical skills in solving real-world problems. In the future, we tend to modify other courses of Software Engineering with CDIO based methodologies for enhancing knowledge building and soft skills in students.

In addition, the knowledge and practical ability of instructors has a deep effect on engineering education, and therefore, universities and colleges must have excellent instructors with professional capacity and practical skills. Several methods are used to improve instructor’s professional and application ability. The first is encouraging instructors to become “double-professionally-titled instructors". The second is regularly carrying out curriculum teaching and research activities, to discuss the method and means of teaching, and allowing instructors to learn the teaching experience from each other. Third is to provide various opportunities and financial support to encourage instructors to attend profession training and all kinds of important education conferences about teaching reform and practice reform, where they can communicate and discuss with colleagues in other universities or in IT Company. Instructors should be encouraged to update their knowledge and teaching materials to keep pace with times.

REFERENCE

Ahmed, F., Capretz, L.F. and Campbell, P., 2012. Evaluating the demand for soft skills in software development. It Professional, 14(1), pp.44-49.

Boehm, B.W. and Sullivan, K.J., 2000, May. Software economics: a roadmap. In Proceedings of the conference on The future of Software engineering (pp. 319-343). ACM.

Crawley, E., Malmqvist, J., Ostlund, S. and Brodeur, D., 2007. Rethinking engineering education. The CDIO Approach, 302, pp.60-62.

Crawley, E.F., 2001. The CDIO syllabus: a statement of goals for undergraduate engineering education (No. s 36). Cambridge, MA: Massachusetts Institute of Technology.

Richter, K. and Dumke, R.R., 2015. Modeling, evaluating, and predicting IT human resources performance. CRC press.

Ghavifekr, S. and Rosdy, W.A.W., 2015. Teaching and learning with technology: Effectiveness of ICT integration in schools. International Journal of Research in Education and Science, 1(2), pp.175-191.

Gustafsson, G., Malmqvist, J., Newman, D.J., Stafström, S. and Wallin, H.P., 2002. Towards a new model for first-year introductory courses in engineering education programmes. NordDesign 2002, Trondheim, Norway, 14-16 August 2002.

Kappelman, L.A., Jones, M.C., Johnson, V., McLean, E.R. and Boonme, K., 2016. Skills for success at different stages of an IT professional's career. Commun. ACM, 59(8), pp.64-70.

Richardson, I. and Delaney, Y., 2009, February. Problem based learning in the software engineering classroom. In 2009 22nd Conference on Software Engineering Education and Training (pp. 174-181). IEEE.

Vo, N.V., Nguyen, D.M. and Ha, N.H., A case study of CDIO implementation in the course of hacking exposed at Duy Tan university. Proceedings of the 13th International CDIO Conference, University of Calgary,Calgary, Canada, June 18-22, 2017.

BIOGRAPHICAL INFORMATION

Nhu-Hang Ha is the Vice Dean of Graduate School, Duy Tan University. She received Ph.D. in Information Management from Yuan Ze University in Taiwan. Her research interests involve Knowledge Management, Project Management, Collaboration, and IT application.

Duc-Man Nguyen is the Acting Dean of the International School, Duy Tan University. His majors are in Software Engineering and Information Systems Management. He has more than 12 years of experience in software development and mentoring for Capstone projects. His interests are in software testing, mobile application testing, and large-scale data processing.

Anand Nayyar is working as Lecturer, Researcher and Scientist in Graduate School, Duy Tan University. A Certified Professional with more than 75+ International Certifications from various IT Giants like Microsoft, CISCO, Oracle, Google. A total of 13+ Years of Academic / Research Experience with more than 300+ Publications in various National, International Conferences cum International Journals (Scopus/ISI/SCI/SCIe) of High Repute. Currently working in the area of WSN, MANET, Swarm Intelligence, Cloud Computing, Big Data, Network Security, Internet of Things (IoT), Machine Learning, Deep Learning, Network Simulation, 5G Communications and many more.

Chia-An Liu is an expert in Human Resource Management. He has just completed his EMBA with a concentration in university and industry collaboration. His main area of interest is the study of HR. His broad experience includes benefits, employee relations, payroll, talent acquisition and management. He has experience in various industries including professional services, technology and construction

Corresponding author

Dr. Nhu-Hang Ha
Graduate School
Duy Tan University
254 Nguyen Van Linh, Danang, Vietnam
+84 707 121 301
hatnhuhang@duytan.edu.vn