Integrating Skills into the Course with the CDIO Approach

Bac LE
University of Science - VNU, Ho Chi Minh City, Vietnam
lhbac@fit.hcmus.edu.vn

Thanh LE
University of Science - VNU, Ho Chi Minh City, Vietnam
lnthanh@fit.hcmus.edu.vn

Tien DINH
University of Science - VNU, Ho Chi Minh City, Vietnam
dbtien@fit.hcmus.edu.vn

ABSTRACT

The skills are not only the course objectives but also the tools which help students acquire knowledge as well as demonstrate their ability to apply knowledge in real life. The paper presents a process to integrate skills into a course based on CDIO framework. First, the skills' key elements are identified. Then, they will be marked as Introduce-Teach-Utilize. Finally, Bloom's taxonomy is used to show the expected level of each element as the learning outcomes of the course. From that, teaching and learning activities are proposed to ensure the student achieve learning outcomes thoroughly. In order to show achievement levels, behaviors and sources of evidence are described in detail. In addition, criteria and standards are also developed for evaluating evidence. For illustration, the paper shows the process of integrating the "Critical Thinking" skill into the "Introduction to Programming" course. The process will show in detail how to integrate skills to a course, which elements of the skill are present as well as levels of achievement by students. Implementing this approach at Faculty of Information Technology (US-VNU) has shown that students can grasp and understand skills, and even adopt them to studying proactively.

Keywords: *Professional knowledge; skills; integration; CDIO approach.*

INTRODUCTION

Universities expect students apply what they learned well in the real life after they graduated. Currently, the teaching and learning activities in University of Science-VNU focus on teaching students on knowledge. Lecturers are focusing much to the content of the course. It is thought that when they are equipped with strong background knowledge, they can apply it easily to real-world problems after graduating. Thus, the school focuses much to the knowledge rather than the skills. However, this worked in the early days in which tasks did not require much creativity and adaptation. In the era of information and technology, things get changed very quickly, almost every day. Employees need to update their knowledge and skills continuously. Products get more complicated and sophisticated. Thus, it requires new knowledge and technology with appropriate skills. Even to maintain or gain the market share, capitalization and profitability, manufacturers are demanding more creativity as well as rapid and continuous innovation. To have highly qualified graduates, knowledge is necessary but not sufficient.

Nowadays industry requires the effort and intellect of not only an individual, but also a team, or even sometime the participation of end-users. Therefore, in addition to knowledge, students need to be equipped with skills such as critical thinking, inquiry and analysis, creative thinking, quantitative literacy, integrated learning, lifelong learning, ethics and professional behaviors.

However, the integration of skills into technical courses is challenging. The problems are what the process of teaching and learning skills is appropriate; and how they are evaluated. In addition, the program should avoid making students overloaded with the school work. Niewoehner (2006) applied a model of critical thinking to classroom. However, this is not designed for the whole course and the assessment methods are not enough. Therefore, this paper focuses on how to bring the specific skill to the course. In particular, the "critical thinking" skill to "introduction to programming" course is chosen to illustrate our approach.

The rest of this paper is organized as follows. Section 1 identifies the key elements of skill which are necessary for the course. Section 2 presents ITU (Introduce/Teach/Utilize) assignment for each key element to ensure their compliance with course objectives or learning outcomes. The level of skills is clarified in section 3. From that, in section 4, the teaching/learning activities are developed to achieve the above objectives. Section 5 defines the sources of evidence which are collected for assessment. Section 6 shares the evaluation criteria which helps students to understand what to do and how much effort to achieve the course objectives. Figure 1 summarizes main steps.

IDENTIFY KEY ELEMENTS

Based on the course objectives, the first step is to identify the key elements that are critical for achieving a specific skill. The elements are presented from low to high, from simple to complex. An example of the elements of the "Critical Thinking" skill include (Figure 2**Error! Reference source not found.**):

- Clearly understanding issues and their scope
- Issue analysis
- Related work of other people
- Hypothesis, evidence, experience
- Discussion and Evaluation
- Conclusion

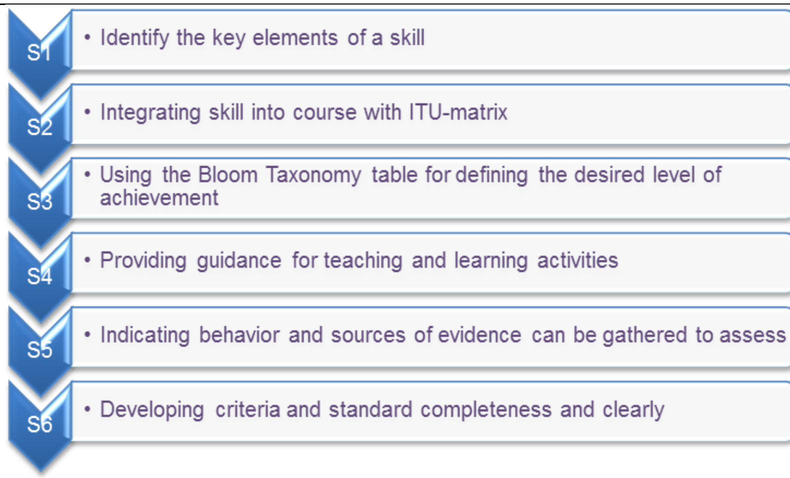

Figure 1: Process of integrating skills into the course

The proposed elements can be based on common factors of the skill. After that, we will select factors which may be used or taught in the course. For example, in introduction to programming course, students are required to become familiar with the different architecture and syntax in programming languages. Each type of syntax only appropriates to some particular problem. For instance, to store and work on the set of objects, students need to decide whether to use many independent variables or just an array of elements of the same data type. Which data types such as integers, real numbers, characters, or strings should be used to declare the object? Therefore, "*comprehension and problem analysis*" element is necessary for this course. Similarly, the other elements are also considered to be put into the course for this skill.

Some detailed learning outcomes of introduction to programming course which are relevance to critical thinking skill include:

- Compare and contrast data types in order to select them in accordance with the specific problem.
- Construct problem solving approaches and describe them in a flow chart or pseudo code form.
- Implement the solution in a programming language based on a given flow chart, pseudo code, or user's description.
- Evaluate the source code on many aspects such that understandable, time consumption, and so on.

It is noticed that students need to be taught about critical thinking to archive the learning outcomes. However, each skill consists of many elements which are chosen differently depending on particular context. Some elements can require a lot of time to teach, the experience of faculty, learning resources as well as learning space. For example, the element "*reflection on the quality of the thinking*" of "*critical thinking*" skill demonstrates a developing sense of oneself as a learner, building on prior experiences to respond to new and challenging contexts. In the case of large class, observing and evaluating this element could be very difficult. Instead, the level of requirement of the element could be reduced or even skipped. Thus, it can be said that the identification of key elements help lecturers target the skill's essential goals that students need to achieve after finishing the course.

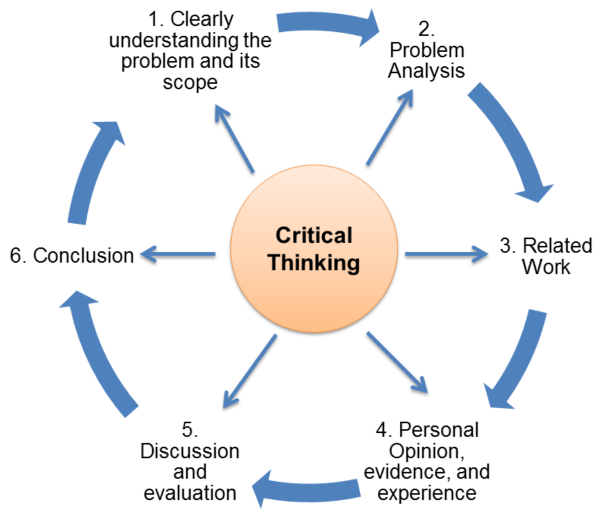

Figure 2: The key elements of critical thinking skill

The elements should be presented from basic to advanced levels sequentially. For example, as shown in Figure 2, the first element begins with understanding the problem that need to analyze or solve. The process is finished with "*conclusion*" element. However, the interesting thing here is that in the last element if the target has not been archived yet, the whole process can repeat from the beginning for another round. This shows the skill development is also similar to the spiral model in which it does not require learners achieve everything in the first round. After having identified the key elements, the next section will describe in detail how to integrate them into the course.

INTEGRATING SKILL'S KEY ELEMENTS

When integrating key elements into courses, it is important to specify the courses' context. For instance, in which year of the program are the courses taught? What are the levels of the students? Based on this, the importance of each factor could be identified in order for the students to achieve them all. Please note that the integration could be specified through the spiral model. That is to say, moving from lower to higher level through years; the skills must be higher and assessed accordingly. As the integration, CDIO (Crawley et al, 2010) suggests the three levels of acquired knowledge: I(Introduced), T(Taught), U(Used). Level I is about the knowledge that should be introduced to students without going deeply in explaining why and how. Usually at this level, lecturers do not need to evaluate students. Level T requires lecturers to teach in a way that focuses on helping students understand thoroughly the content so that they can use it later in practice. The third one, level U, assumes students already understand the knowledge; as a result, the lecturers expect students to be able to apply or to enhance the ability to apply in other areas. In the same course, different levels can be applied to each element. For example, marking I/T means that the lecturers will both introduce and teach that element to students. It might be the case that students are at initial stages of knowing and understanding new concepts. For example, in integrating key elements of "critical thinking" into "introduction to programming", we mark ITU for each element as shown in Table 1.

Table 1: Integrating key elements with ITU-matrix

	1. Under-standing	**2. Analysis**	**3. Related Work**	**4. Hypo-thesis**	**5. Discussion Evaluation**	**6. Conclu-sion**
Introduction to Programming	**T**	**T**	**I**	**I,T**	**I**	**I**

In University of Science-VNU, because the "introduction to programming" and "soft skill" courses are taught concurrently, understanding and analysis elements have been introduced in "soft skill" course (level I). Therefore, the "introduction to programming" course will focus on teaching students how to understand a problem; which factors are required; and which factors are important. Thus, T level is marked to the first and second elements. For example:

Let writing the computer program to calculate the population of Ho Chi Minh City in year X. Suppose that the City's population is 9 million people in 2011. The average annual population growth rate is 2.14%.

Students need to understand that the population only provided in 2011 but not before. They need to ensure that the program does not raise any error when a user inputs a year sometime before 2011. In addition, the population is a positive integer, but growth rate is a real number. Therefore, how to understand and analyze the problem needs teaching by lecturers when students are new to programming. However, with the third element, since this course does not require students to find related work, level I is marked for this element. It means the concepts will be just introduced freshmen, but not require students to understand or be able to apply. With the fourth element, since students have not been heard or known it before, both I and T are assigned to this element. In that case, lecturer will firstly introduce and then instruct methods for students to make assumption or hypothesis, offering solutions, doing experience, and explaining results as well.

DEFINING THE DESIRED LEVELS

After each I, T, U was determined on key elements, we need to clarify the desired level in detail. Specifically, Level Objective and Group Objective are identified. This step is often referred to as *Learning Outcome Map*.

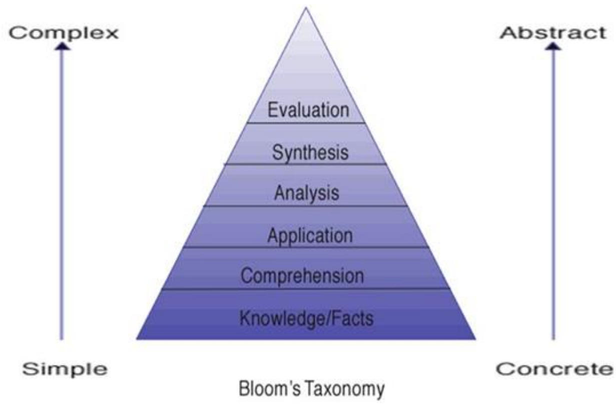

Figure 3: Levels in Bloom's Taxonomy (Overbaugh et al, 2013)

In order to achieve that, Bloom's Taxonomy of Education Objectives is used. Bloom's taxonomy may be used to provide the description of "what students should do if they are to achieve the intended outcome", for example, know, comprehend, apply, analyze, synthesize, or evaluate. The *introduce* (I), *teach* (T), *use* (U) categories may be mapped by the Bloom's taxonomy. When the learning is *introduced* may imply that students will know facts, concepts, principles or conventions. Thus, level I typically correspond to *knowledge* level in Bloom's taxonomy. *Comprehension*, the next level, requires students to understanding what was learned. It means lecturers must teach (T) them. Similarly, the rest such as *application*, *analysis*, *synthesis* and *evaluation* implies level U in ITU-matrix. Figure 3 shows the Bloom's taxonomy. It requires more efforts from students for higher levels of understanding.

In the above Level Objectives, we also need to list groups of learning outcomes organized around the Bloom's Taxonomy levels. For example, if the learning outcome is at the *knowledge* (*level*), the student will be able to *list* (*group*) or *describe* (*group*) learned materials. Or, if the learning outcome is at the *comprehension* (*level*), the student will be able to *explain* (*group*) or *translate* (*group*) relevant terms, concepts, etc. This helps lecturer to identify the outcomes clearly. Level Objectives and Group Objectives are listed to map learning outcomes (Crawley, 2001). Table 2 illustrates Level and Group Objectives which used for integrating critical thinking into the course.

Table 2: Learning Outcome Map for critical thinking skill

Course: Introduction to Programming			
Key Elements	**I/T/U**	**LO (level)**	**LO (Group)**
1. Understanding	T	Comprehension	Locate and Classify Explain Translate
	U	Application	Prepare Use
2. Analysis	T	Comprehension	Locate and Classify Explain
3. Related Work	I	Knowledge	Recognize List
4. Hypothesis	I	Knowledge	List Describe
	T	Comprehension	Translate
5. Discussion and Evaluation	I	Knowledge	Describe Match
6. Conclusion	I	Knowledge	Recognize List

According to Table 2, level T is marked for *understanding* an element. Thus, *comprehension* is selected for the element objective. However, comprehension needs to be identified in more detail at the group objective. For instance, students can *name* or *locate* a number of concepts such as eight queens' puzzle, Fibonacci number, Tower of Hanoi etc. Furthermore, they can *classify* requirements such as making out the electric bills, displaying moving texts, etc. In addition, students can also *explain* and *translate* the concepts such as divisor, Euclidean division, integer, floating point number in computer science. They can convert mathematical formula into the calculations on array or pointers in programming languages. If level U is selected, it means *application* level is chosen. At group objective, students are able to *prepare* and *use* learned methods. For instance, students can finish programming assignments with pointers and linked lists. The process sequence is similar to the previous for the remaining elements (Table 2).

GUIDING FOR TEACHING AND LEARNING ACTIVITIES

This section will address the question of whether the teaching and learning activities help students achieve the learning outcomes. Based on the CDIO approach (Crawley et al, 2010), teaching and learning activities need to be closely linked to both desired student learning outcomes and learning assessment. For example, lecturer designs significant learning experiences (Fink, 2003). The main purpose is to encourage students to participate actively and be proactive in comprehending and applying what they learned. Lecturers act as counselors, instructors, supporters, or even friends by which help students to achieve the course objectives. However, note that the teaching and learning activities must be consistent with defined objectives. This step is often referred to *Teaching/Learning Map*.

The activities can be divided into four main groups (Svinick et al, 1987):
- *Concrete Experience*
- *Reflective Observation*
- *Abstract Conceptualization*
- *Active Experimentation*

Lecturers can start with any group depending on the classroom context. The diagram in Figure 4 illustrates this group.

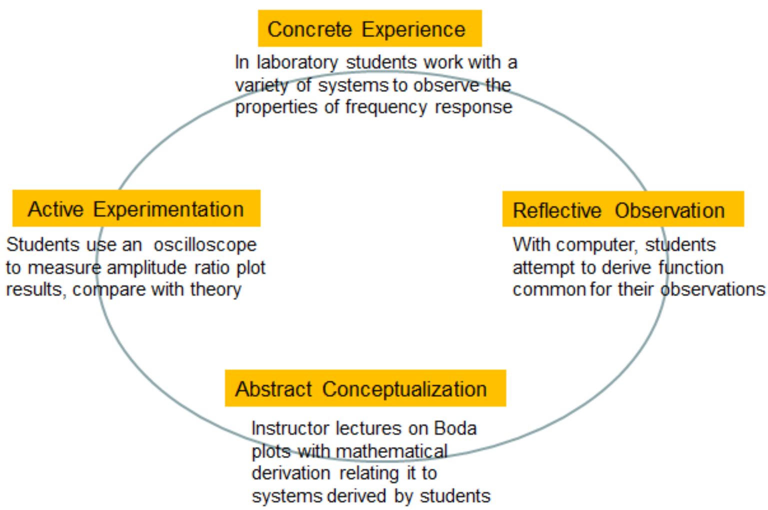

Figure 4: Teaching and learning activities (Svinick et al, 1987)

In Table 3, we propose the teaching/learning activities to achieve the objectives for critical thinking skills. For example, for the *analysis* element, lecturers can give some case studies and analyze them. After that, students brainstorm as well as discuss on some issues. For the *conclusion* element, student will write reports describing conclusions about low complexity projects at home after the lecturer summarizes main points. Each teaching and learning activity can spread across more than one element.

Specifically, to achieve the *match* level in the fifth element, lecturers can make the students think about the advantages and disadvantages of each proposed solution. For example, lecturers might ask students to manage customer's information include the name (string type), identity card number (string type), check-in and check-out date (date type) in computer programs. Students can use arrays, structures or linked list to store and implement the information. However, each approach or data structure has strengths and weaknesses. For instance, lecturers can ask questions like this "Please consider the deletion process of each method and give comments", "What the errors are usually caused when memory is allocated?". As a result, students can learn a lot by answering those questions and give comments.

Table 3: Teaching and learning activities for critical thinking skill

Course: Introduction to Programming			
Key Elements	**LO (Group)**	**Teacher's Activities**	**Student's Activities**
1. Understanding	Locate and Classify	Demonstration Lecture	Following the step-by-step tutorials and answer questions.
	Prepare Use	Assign some low-complexity projects	Solving projects
2. Analysis	Explain	Introducing some case study and analyzing.	Brainstorming Discussion
3. Related Work	List	Give reference books	Guided Reading
4. Hypothesis	Translate	Demonstration	Role-play exercise
5. Discussion and Evaluation	Match	Give thought questions	Comparison and give comments
6. Conclusion	Describe	Summarize the main points and give open question	Write report

INDICATING BEHAVIORS AND SOURCES OF EVIDENCE

When Level and Group Objectives have been identified, we evaluate how far students achieve intended objectives. The session is to describe the *behaviors* that students will perform in order to demonstrate learning achievement. This involves transforming learning outcomes into behavioral objectives. The behaviors are verbs appropriate to defined Level and Group Objectives in the Bloom's Taxonomy (Overbaugh et al, 2013). For example, if Group Objective is selected as *explain*, students should be able to *discuss* to make the interpretation of a given problem or *give examples* as well as *summarize*, *report*, etc.

Next, lecturers gather information to assess students. However, what information should be gathered? What evidence can be collected to document achievements? Therefore, the list of evidence is mentioned for assessing student learning. There are many sources of evidence, such as short essays, assignments, reports, and so on (Biggs et al, 2011). Please note that, the same evidence can be used for to show the achievement of different skills/behaviors. For example, *thought questions* can be used to illustrate the achievement of "discussion" and "arrangement". Table 4 presents behaviors and sources of evidence corresponding to elements.

Table 4: Behaviors and source of evidence

Course: Introduction to Programming			
Key Elements	**LO (Group)**	**Behavior**	**Source of evidence**
1. Understanding	Locate and Classify	Identify Arrange	Thought question Peer Observation
	Use	Apply	Homework
2. Analysis	Explain	Discuss Give examples	Thought question Group Discussion
3. Related Work	List	Outline	Report Question and answer
4. Hypothesis	Translate	Interpret Convert	Group Discussion
5. Discussion and Evaluation	Match	Point out	Oral Presentation
6. Conclusion	Describe	Summarize	Report Homework Essay

In Table 4, an element is chosen to illustrate the approach. For example, how to know students are able to *explain* their analysis in the 2nd element. The student can be given programming examples and asked to analyze them. Students might be asked to present it in front of other members or discuss in group. Meanwhile, lecturers observe the entire process; gather evidence such as noting important points, taking pictures when the students work. Or, lecturers can give some thought questions to check key points. For instance, "please tell me when the program gets to a failure state if the array is allocated statically?" The student must analyze the fail states and explain the connection between fails and the reasons. Based on that, lecturers can see the skills of organizing and synthesizing evidence from the students.

DEVELOPING CRITERIA AND STANDARDS

The final step is to identify criteria for assessing gathered evidences. Based on the criteria, the performance level is defined, such as capstones, milestones, and benchmarks. Each one is scored as a number. However, the criteria and the rubrics should be defined clearly for the students to understand and to be able to do a self-assessment.

For example, criteria are developed for evaluating *understanding* element including: describing clearly and comprehensively, give necessary information to show a deep understanding. With high competence, students are required to ensure the sufficiency and even more than expectation. In *understanding* element, if a student wants to achieve the capstone level, he/she must consider problem carefully, describe it clearly and comprehensibly, and deliver all relevant information necessary to show a deep understanding. However, if the student considers the problem without clarification or description details, he only reaches the benchmark level. Table 5 is used for illustration of the skill.

Table 5.Criteria for critical thinking skill

	Capstone	**Milestones**		**Benchmark**
	1	2	3	4
Analysis	Organizes and synthesizes evidence to reveal insightful patterns, differences, or similarities related to focus.	Organizes evidence to reveal important patterns, differences, or similarities related to focus.	Organizes evidence, but the organization is not effective in revealing important patterns, differences, or similarities.	Lists evidence, but it is not organized and/or is unrelated to focus.

In addition, the course must show the minimum passing level to ensure the desired quality. This is called the *passing standard* of achievement. For example, students with scores of 5 (out of 10) or more will pass. Otherwise, he/she fails.

CONCLUSION

The paper presents a process to integrate skills into the course based on CDIO approach. The process begins with identifying key elements of a specific skill. Next, ITU is attached for each element based on course's context. After that, Bloom's Taxonomy is used for mapping ITU to the learning outcomes. Then, teaching and learning activities are proposed to help students to achieve learning outcomes. In order to specify achievement levels, behaviors and sources of evidence are noted down using verbs in Bloom's Taxonomy. Finally, criteria and standards are developed so that student can be able to understand clearly when he/she passes or fails; and what he/she should do to achieve desired objectives. The advantage of this approach is that it helps lecturers and students identify easily the content to be learned and what students should do if they are to achieve the intended outcome.

REFERENCES

Biggs, J., & Tang, G. (2011). *Teaching for Quality Learning at University*. Buckingham, England: Open University Press/McGraw Hill.

Crawley, E.F., Malmqvist J., Ostlund, S., & Brodeur, D. (2010). *Rethinking Engineering Education: The CDIO Approach*. New York: Springer.

Crawley, E.F. (2001). *The CDIO Syllabus: A Statement of Goals of Undergraduate Engineering Education*. MIT/Cambridge, Massachusetts.

Fink, L.D. (2003). *Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses*. San Francisco: Jossey-Bass.

Niewoehner, R. J. (2006). Applying a critical thinking model for engineering education. *World Transactions on Engineering and Technology Education*, 5(2), 341.

Overbaugh, R.C., & Schultz, L. (2013). *Bloom's Taxonomy*. Retrieved January 10, 2013, from http://ww2.odu.edu/educ/roverbau/Bloom/blooms_taxonomy.htm

Svinick, M. D., & Dixon, N. M. (1987). *The Kolb Model Modified for Classroom Activities* (pp.141-146). Taylor & Francis.