INVITED ARTICLE

A FIRST COURSE IN SOFTWARE ENGINEERING METHODS AND THEORY

UN CURSO INICIAL SOBRE TEORÍA Y MÉTODOS DE LA INGENIERÍA DE SOFTWARE

CARLOS ZAPATA
Ph.D., Profesor Asociado, Facultad de Minas, Universidad Nacional de Colombia, sede Medellín, cmzapata@unal.edu.co

IVAR JACOBSON
Ph.D., Chairman of Ivar Jacobson International, ivar@ivarjacobson.com

Received for review December 10th, 2013, accepted February 04th, 2014, final version February, 10th, 2014

ABSTRACT: Even though the software engineering curriculum has been designed many times, some problems still remain: the gap between academy and industry, the failure to continuously update the courses, the difficulties for combining theory and practice, and the lack of a sound, widely accepted theoretical basis. The SEMAT (Software Engineering Methods and Theory) initiative has been proposed for addressing some of the aforementioned problems. Based on the main ideas related to SEMAT, in this paper we propose a first course that introduces students to the main issues about SEMAT. This course is planned to be included in the System and Informatics Engineering Program belonging to the Universidad Nacional de Colombia, Medellin Branch. Also, we discuss the way in which this course addresses the previously diagnosed problems.

Key words: SEMAT, curriculum design, software industry, software academy, software community.

RESUMEN: Aunque desde hace años se habla del diseño curricular de la ingeniería de software, algunos problemas aún subsisten: la brecha existente entre la academia y la industria del software, los problemas para mantener actualizados los cursos, las dificultades para combinar la teoría con la práctica y la carencia de una sonora y ampliamente aceptada base teórica para la ingeniería de software. La iniciativa SEMAT (nombrada así por el acrónimo en inglés de Teoría y Métodos de la Ingeniería de Software) se propuso para solucionar algunos de los problemas en mención. Tomando como base las ideas relacionadas con SEMAT, en este artículo se propone un primer curso que suministra a los estudiantes una introducción a los principales tópicos de SEMAT. Se planea incluir este curso en el programa de Ingeniería de Sistemas e Informática de la Universidad Nacional de Colombia, Sede Medellín. También, se discute la manera en que este curso contribuye a solucionar los problemas que se diagnosticaron previamente.

Palabras clave: SEMAT, diseño curricular, industria del software, academia del software, comunidad del software.

1. INTRODUCTION

Since the early work of Farley [1], several authors have been working on curriculum design of software engineering courses [2–5]. Some authors are focused on complete curricula [1, 2] while others are focused on courses for meeting the industry needs [3–5]. However, all of them recognize the same remaining problems in the software engineering curriculum: (i) the growing gap among the software engineering courses created by academy and industry needs; (ii) the failure experienced by professors in order to keep up-to-date the contents of the courses, since there are fads emerging day-by-day related to software engineering methods and practices; (iii) the actual difficulties for offering practices matching the theories related to this field of knowledge; and (iv) the lack of a theoretical basis for the software engineering, even though there is a strong work on formal methods and conceptualization.

Jacobson et al. [6] proposed the SEMAT (Software Engineering Methods and Theory) initiative as a way to deal with several problems related to software engineering as a discipline by re-founding it. So, the SEMAT community has focused on two major goals: (i) finding a kernel of widely-agreed elements, and, (ii) defining a solid theoretical basis of the software engineering.

The main ideas of the SEMAT community lead us to propose a first course in software engineering methods and theory to be included in the System and Informatics Engineering program belonging to the Universidad Nacional de Colombia, Medellín Branch. The course syllabus comprises several activities, ranging from lectures—some of them held by international guests—to active strategies such as forums, games, and practical projects. Also, some discussion is promoted in order to show the way in which we can address the aforementioned problems related to software industry and software academy, since they are closely linked to the progress of this discipline.

The reminder of this paper is organized as follows: in Section 2 we present some background related to the software engineering curriculum. The main description of the SEMAT initiative is presented in Section 3. Then, in Section 4 we propose a first course in software engineering methods and theory and we promote some discussion about the way in which some problems are addressed by this course. Finally, we present conclusions and future work.

2. SOFTWARE ENGINEERING CURRICULUM BACKGROUND

Farley [1] has probably the first attempt to define a software engineering undergraduate program. Such a program was designed for providing some core elements of the software engineering, but—as the author specifically recognized—these elements did not provide the adequate training for a software engineer. Mead [2] gives some insight about the further development of the software engineering curriculum. She also points out the strong urge for correlating the software engineering curriculum to the industrial practice.

Ludewig and Reißing [3] discuss the importance of theoretical courses as a way to provide tools to the software engineer for addressing real problems. Amiri et al. [4] make a survey to several managers of the software industry to the extent of determining the appropriateness of the software engineering courses. Moreno et al. [5] make a similar study about the software engineering courses versus the needs of the industry.

Being related to either curriculum as a whole or isolated software engineering courses, the previously reviewed work exhibits a consensus about some difficulties experienced by the software engineering education nowadays. Further information about some common problems is provided as follows:

Industry vs. Academy. The software Industry is demanding a different software engineer than the one promoted by the academic curriculum. In fact, some of the most demanded skills are usually far beyond the software engineering curricula. Some of the software industry managers claim the strong need to re-train the software engineering newcomers when they arrive from universities.

Courses vs. New methods. New software engineering methods are emerging day-by-day, creating new trends to be followed by practitioners. This fact poses a big challenge upon the software engineering course designers: how to keep up-to-date the course contents when several trends are coming from the academic world and they are being adopted by the software industry? How software engineering courses can evolve to follow the new trends in methods and practices?

Theory vs. Practice. Lectures are common ways of teaching software engineering and practice is usually limited to “toy” projects in some software engineering courses. Software industry needs practical, skillful engineers. How can we train practical professionals when the education is mainly theoretical?

Software Engineering Theoretical Basis. Methods and practices are emerging at a fast rate, since many research groups are promoting new ideas for dealing with the software engineering problems. However, we can argue against the novelty of such ideas, because many of them use previously known concepts. Even though some effort is devoted to define bodies of knowledge and glossaries about software engineering, we still lack a sound, widely-agreed theoretical basis for software engineering.

The SEMAT initiative is intended to deal with the aforementioned problems, as described in the following Section.

3. THE SEMAT INITIATIVE

Jacobson et al. [6] propose the SEMAT initiative for refunding the software engineering by defining a theoretical basis—a small set of widely-agreed elements. Such elements are useful for defining past, present, and future methods, so the practices included on them can be reused. As a result, some practices seem to belong to obsolete methods and they can be incorporated into modern ones, as software engineering evolves. Theoretical basis is gathered into the so-called SEMAT kernel—the Essence of software engineering. The kernel is represented into a language with a limited number of elements (called alphas, see Fig. 1). The alphas can be used for assessing the health and progress of a software endeavor

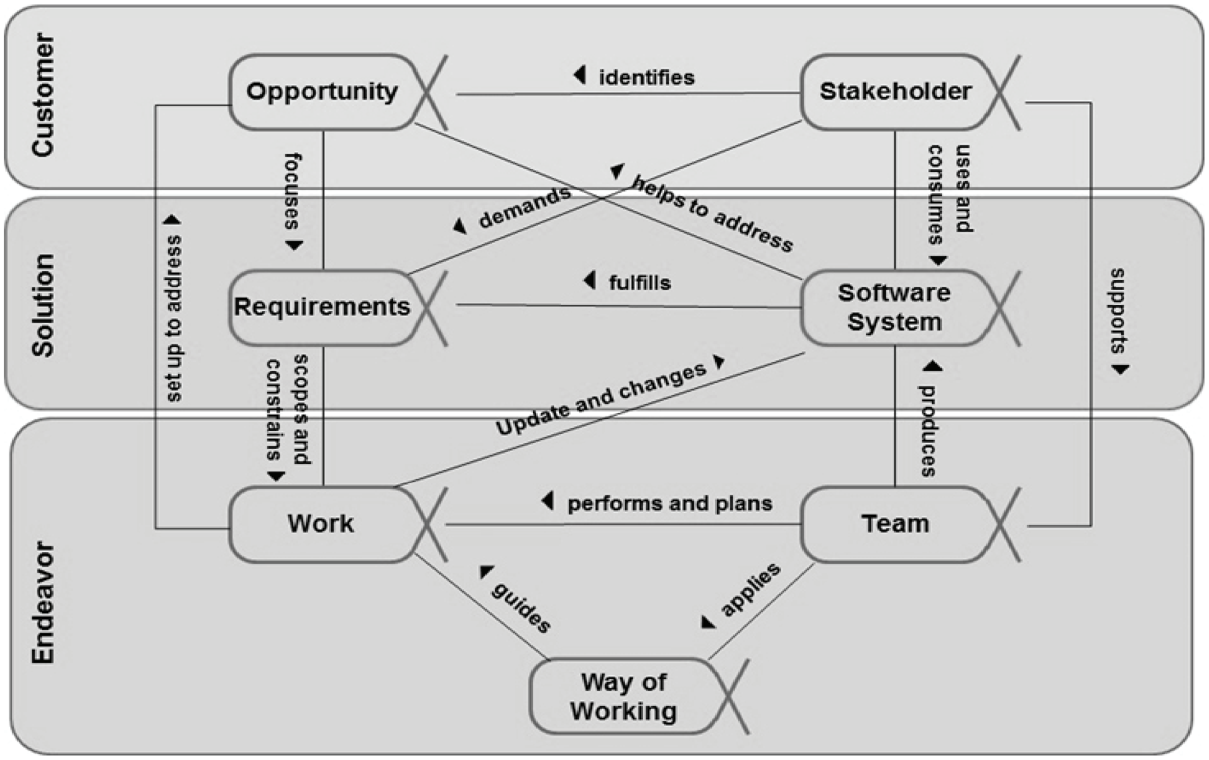
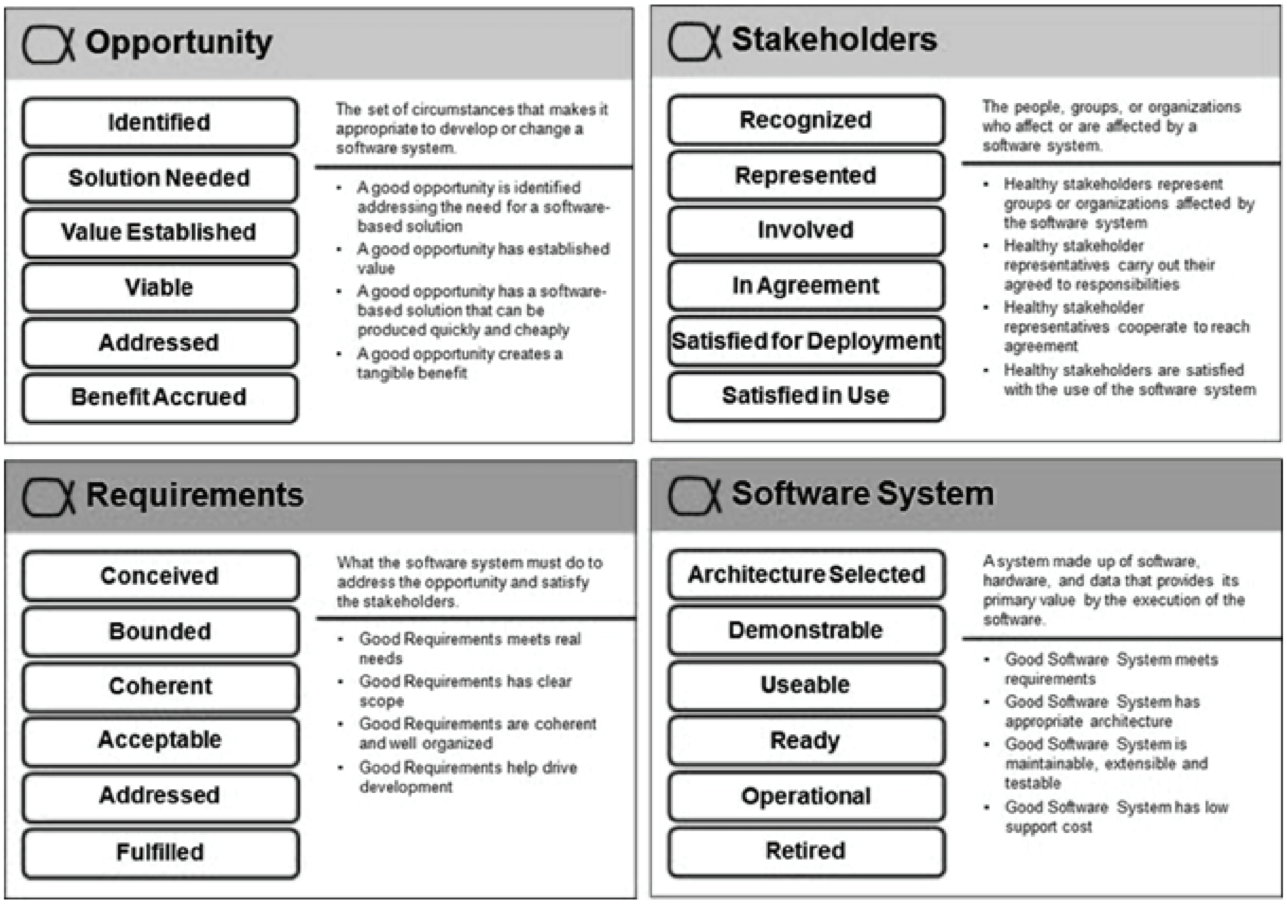


Figure 1. Alphas of the SEMAT kernel [6].

Health and progress can be estimated by using the alpha states. Some cards can be used in the SEMAT kernel for (i) describing alphas and possible states (see an example in Fig. 2), and (ii) linking alpha states to checklists for the sake of assessing their compliance (see an example in Fig. 3). Some other elements complement the theoretical basis of the software engineering represented by the SEMAT kernel. Some of them are: areas of concern, activity spaces, activities, competencies, and work products.

In this paper we propose a structure based on the so-called pre-conceptual schemas [7] for representing the SEMAT kernel elements and their relationships. In such schemas, nouns are represented by rectangles and verbs are represented by ovals. Thin arrows are used to connect nouns and verbs in such a way they express phrases, e. g. "method has practice" and "methodologist develops kernel." Thick arrows are used to express cause-and-effect relationships, e. g. "methodologist uses kernel, then methodologist describes practice." Finally, dotted boxes are used to express possible values of the nouns linked to them, e. g. "customer, solution, and endeavor are possible values of area of concern."

Fig. 4 depicts the kernel structure. In this figure, elements shown in blue were added to the schema to the extent of making it more readable and logical.



The set of circumstances that makes it appropriate to develop or change a software system.

- A good opportunity is identified addressing the need for a software-based solution
- A good opportunity has established value
- A good opportunity has a software-based solution that can be produced quickly and cheaply
- A good opportunity creates a tangible benefit

The people, groups, or organizations who affect or are affected by a software system.

- Healthy stakeholders represent groups or organizations affected by the software system
- Healthy stakeholder representatives carry out their agreed to responsibilities
- Healthy stakeholder representatives cooperate to reach agreement
- Healthy stakeholders are satisfied with the use of the software system

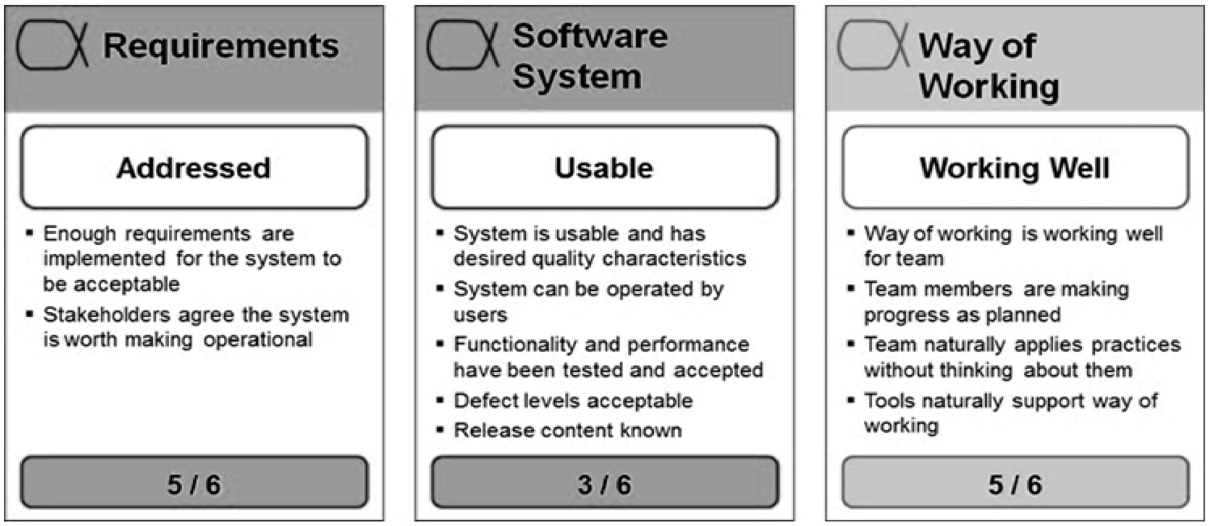
What the software system must do to address the opportunity and satisfy the stakeholders.

- Good Requirements meets real needs
- Good Requirements has clear scope
- Good Requirements are coherent and well organized
- Good Requirements help drive development

A system made up of software, hardware, and data that provides its primary value by the execution of the software.

- Good Software System meets requirements
- Good Software System has appropriate architecture
- Good Software System is maintainable, extensible and testable
- Good Software System has low support cost

Figure 2. Examples of alphas and their states [6].



- Enough requirements are implemented for the system to be acceptable
- Stakeholders agree the system is worth making operational

- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

- Way of working is working well for team
- Team members are making progress as planned
- Team naturally applies practices without thinking about them
- Tools naturally support way of working

Figure 3. Examples of alpha states and checklists [6].

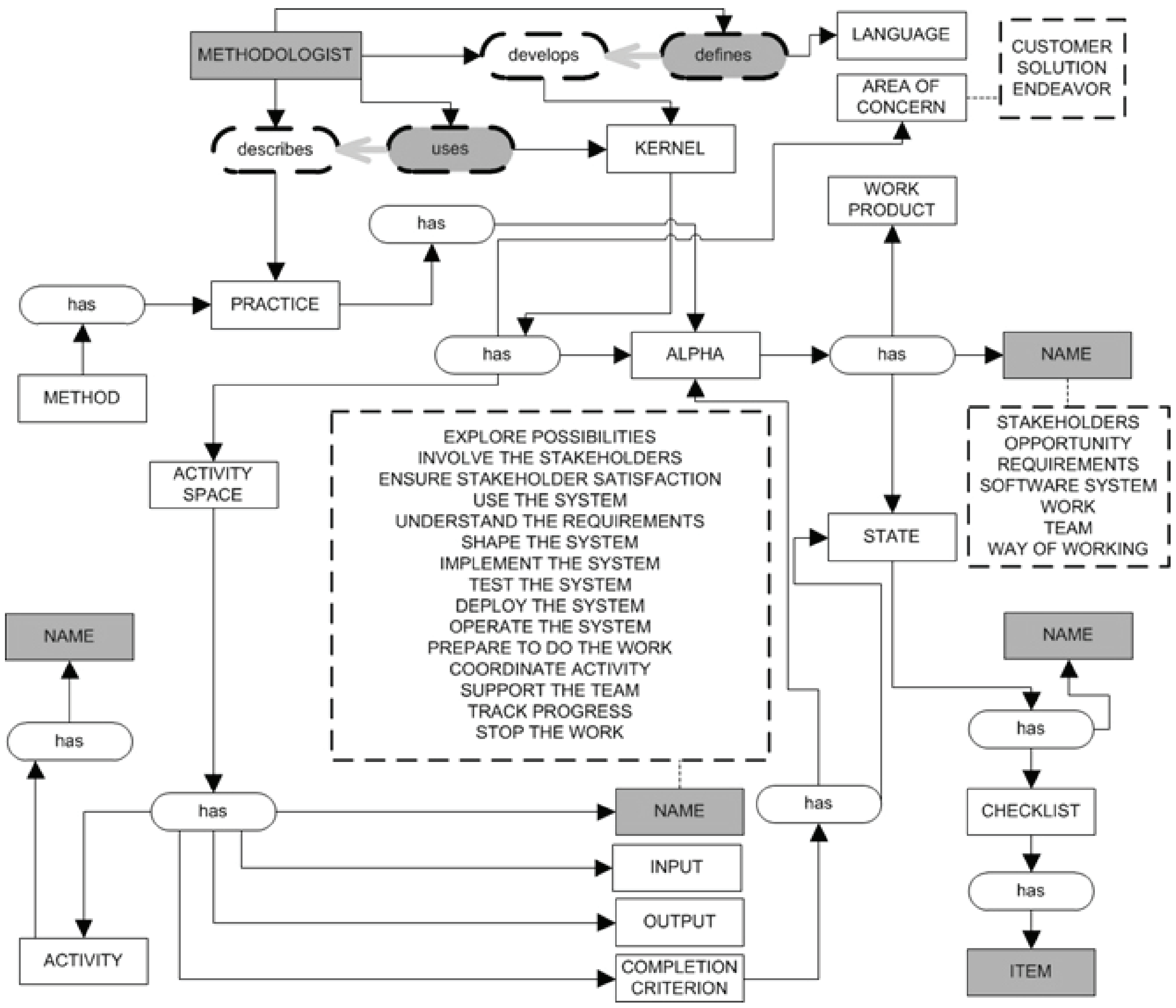

METHODOLOGIST
develops
defines
LANGUAGE
AREA OF CONCERN
CUSTOMER SOLUTION ENDEAVOR
describes
uses
KERNEL
has
WORK PRODUCT
has
PRACTICE
METHOD
has
ALPHA
has
NAME
STAKEHOLDERS OPPORTUNITY REQUIREMENTS SOFTWARE SYSTEM WORK TEAM WAY OF WORKING
ACTIVITY SPACE
EXPLORE POSSIBILITIES INVOLVE THE STAKEHOLDERS ENSURE STAKEHOLDER SATISFACTION USE THE SYSTEM UNDERSTAND THE REQUIREMENTS SHAPE THE SYSTEM IMPLEMENT THE SYSTEM TEST THE SYSTEM DEPLOY THE SYSTEM OPERATE THE SYSTEM PREPARE TO DO THE WORK COORDINATE ACTIVITY SUPPORT THE TEAM TRACK PROGRESS STOP THE WORK
STATE
NAME
has
NAME
has
CHECKLIST
has
has
NAME
INPUT
OUTPUT
COMPLETION CRITERION
ACTIVITY
has
ITEM

Figure 4. A pre-conceptual schema for representing the SEMAT kernel elements and their relationships. Source: the authors.

4. A FIRST COURSE IN SEMAT

SEMAT kernel is intended to address some of the problems exposed in Section 2, but a course related to the kernel is needed for addressing the four problems. In this Section we propose the curricular design of such a course and we discuss the way in which the problems are addressed.

4.1. Curricular Design

General information about the course is the following:

Code: 3009583
Name: Software Engineering Methods and Theory
Program: Systems and Informatics Engineering
Department: Computer and Decision Sciences
Level: Undergraduate
Typology: Free choice
Weekly in-classroom activity: 4 hours
Weekly outside-classroom activity: 5 hours
Weeks per semester: 16
Total activity: 144 hours
Credits: 3

General Objective: Defining and specifying the elements underlying any software engineering method and practice

Specific Objectives: (i) Establishing the cross-cutting elements to explaining software engineering methods and practices in terms of the SEMAT kernel; (ii) generating specific skills for representing any method and its practices in the SEMAT kernel; (iii) specifying any method and its practices in terms of the SEMAT kernel formal language; and (iv) recognizing and assessing tools used for creating representations in the SEMAT kernel

Methodology and Assessment: (i) Lectures; (ii) serialized practical project; (iii) forums; and (iv) experience-based games

The contents of the course are the following:

A. Initial information about software engineering general theories

A.1. Motivation

A.2. General Problems related to software engineering

A.3. Need for a software engineering theory

B. Basic elements of the software engineering kernel

B.1. Alphas and card-based representation

B.2. Activity spaces

B.3. Methods and practices

B.4. Competencies

B.5. Work products

C. Advanced elements of the software engineering kernel

C.1. Patterns

C.2. Resources

C.3. Detail levels

C.4. Competency levels

C.5. Separation of concerns

C.6. Tools for working with the kernel

D. Formal representation of the software engineering kernel

D.1. Introduction

D.2. Kernel meta-model

D.3. Kernel text-based specification

D.4. Object diagrams and executable pre-conceptual schemas

The course syllabus is included in Table 1. Several remarks are needed:

- Each numbered class (1 to 32) is a 2-hour class and has the issue to be covered, the contents related to such issue and the assessment items.

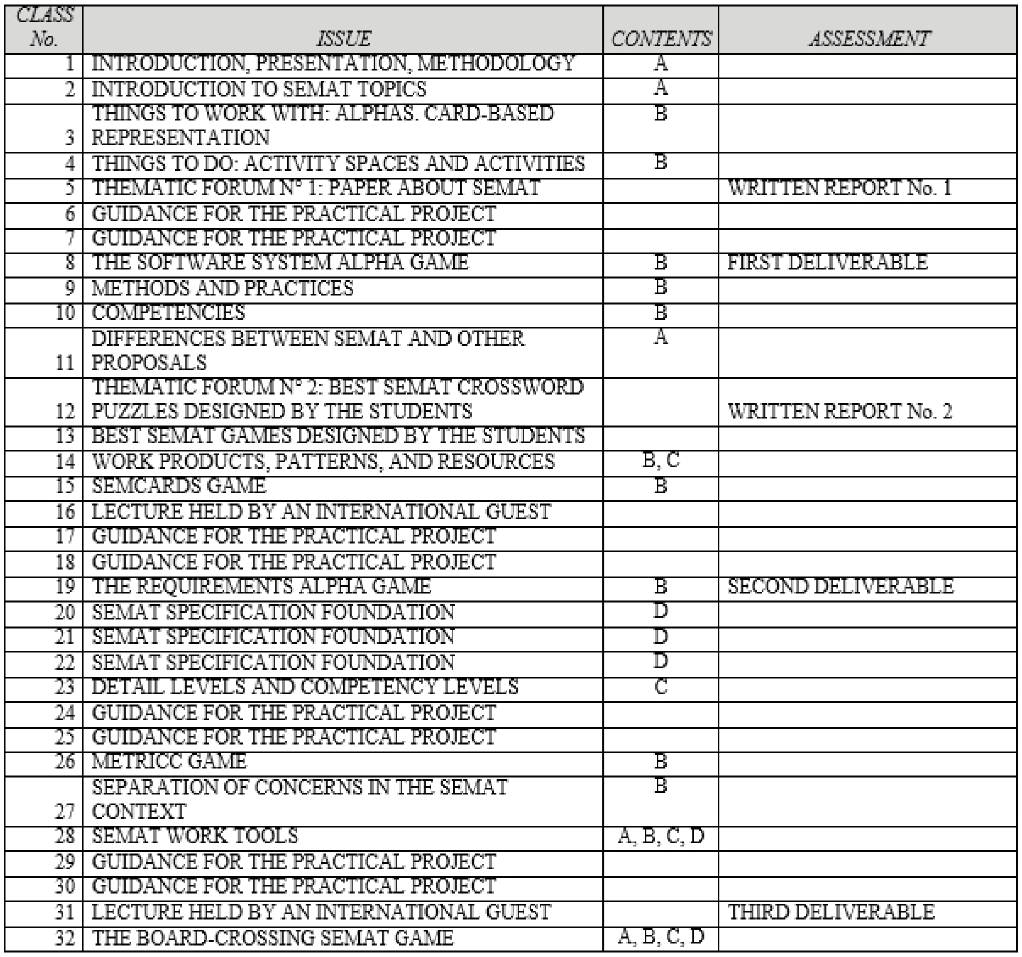
- Almost all the lectures should be held by the course professor, with the exception of class No. 16 and 31, which should be held by an international guest. In this case, we try to take advantage of the excellent relationships we have with the SEMAT Community and invite some international SEMAT researchers to hold a lecture via videoconference or keynote—in such a case the international guest is visiting our University.

- The effort to develop the practical project is intended to be guided in the classroom, with the participation of all the students. Six classes are reserved for this purpose. The topic of the practical project is related to the representation of a method by using the SEMAT kernel.

- The first forum is devoted to practicing the initial skills needed for developing the practical project. The second forum is devoted to activities developed by the students by using active learning: the design of a crossword puzzle with the main concepts of SEMAT and the design of an experience-based game by studying some of the SEMAT concepts.

- Five pre-designed games will be played during the course: the software system alpha game (see Fig. 5), the requirements alpha game (see Fig. 6), the SemCards game (see Fig. 7), the MetricC game (see Fig. 8), and the SEMAT board-crossing game (see Fig. 9). All of these games were previously designed and tested with several groups of heterogeneous people, mostly during Conferences, e. g. the Colombian Computing Conference [8], the Latin American Conference on Informatics [9], and the International Conference on Science and Technology for the Risk Management and the Climate Change Adaptation.

Table 1. SEMAT course syllabus. Source: the authors.



CLASS No.	*ISSUE*	*CONTENTS*	*ASSESSMENT*
1	INTRODUCTION, PRESENTATION, METHODOLOGY	A	
2	INTRODUCTION TO SEMAT TOPICS	A	
3	THINGS TO WORK WITH: ALPHAS. CARD-BASED REPRESENTATION	B	
4	THINGS TO DO: ACTIVITY SPACES AND ACTIVITIES	B	
5	THEMATIC FORUM N° 1: PAPER ABOUT SEMAT		WRITTEN REPORT No. 1
6	GUIDANCE FOR THE PRACTICAL PROJECT		
7	GUIDANCE FOR THE PRACTICAL PROJECT		
8	THE SOFTWARE SYSTEM ALPHA GAME	B	FIRST DELIVERABLE
9	METHODS AND PRACTICES	B	
10	COMPETENCIES	B	
11	DIFFERENCES BETWEEN SEMAT AND OTHER PROPOSALS	A	
12	THEMATIC FORUM N° 2: BEST SEMAT CROSSWORD PUZZLES DESIGNED BY THE STUDENTS		WRITTEN REPORT No. 2
13	BEST SEMAT GAMES DESIGNED BY THE STUDENTS		
14	WORK PRODUCTS, PATTERNS, AND RESOURCES	B, C	
15	SEMCARDS GAME	B	
16	LECTURE HELD BY AN INTERNATIONAL GUEST		
17	GUIDANCE FOR THE PRACTICAL PROJECT		
18	GUIDANCE FOR THE PRACTICAL PROJECT		
19	THE REQUIREMENTS ALPHA GAME	B	SECOND DELIVERABLE
20	SEMAT SPECIFICATION FOUNDATION	D	
21	SEMAT SPECIFICATION FOUNDATION	D	
22	SEMAT SPECIFICATION FOUNDATION	D	
23	DETAIL LEVELS AND COMPETENCY LEVELS	C	
24	GUIDANCE FOR THE PRACTICAL PROJECT		
25	GUIDANCE FOR THE PRACTICAL PROJECT		
26	METRICC GAME	B	
27	SEPARATION OF CONCERNS IN THE SEMAT CONTEXT	B	
28	SEMAT WORK TOOLS	A, B, C, D	
29	GUIDANCE FOR THE PRACTICAL PROJECT		
30	GUIDANCE FOR THE PRACTICAL PROJECT		
31	LECTURE HELD BY AN INTERNATIONAL GUEST		THIRD DELIVERABLE
32	THE BOARD-CROSSING SEMAT GAME	A, B, C, D	

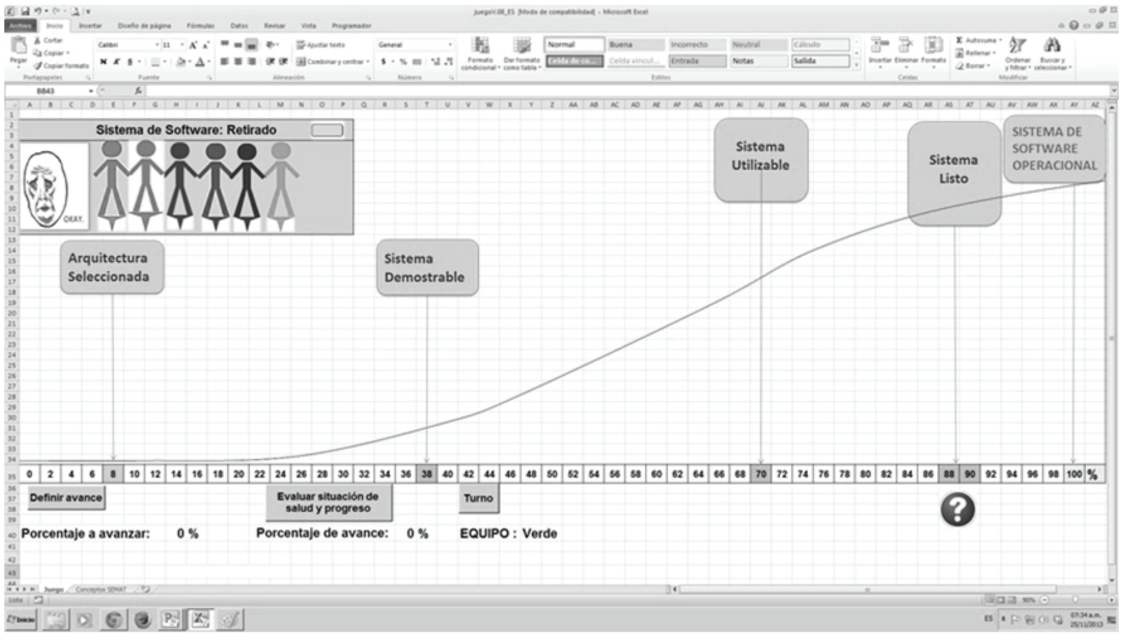
4.2. Discussion

The four problems presented in Section 2 can be addressed by the SEMAT course we propose in this paper. The reasons are the following:

Industry vs. Academy. While the SEMAT initiative is advancing in the world, several discussions between practitioners and academics have been promoted in order to standardize the terminology related to software engineering. The proposed course is a direct result of such discussions and, consequently, it is intended to close the gap between the software engineering curriculum and the industrial training needed by software engineers. Also, two international guests will hold lectures to keep students informed about what is happening in the software industry. Since we can promote such a kind of lectures, we are also working to close the aforementioned gap.

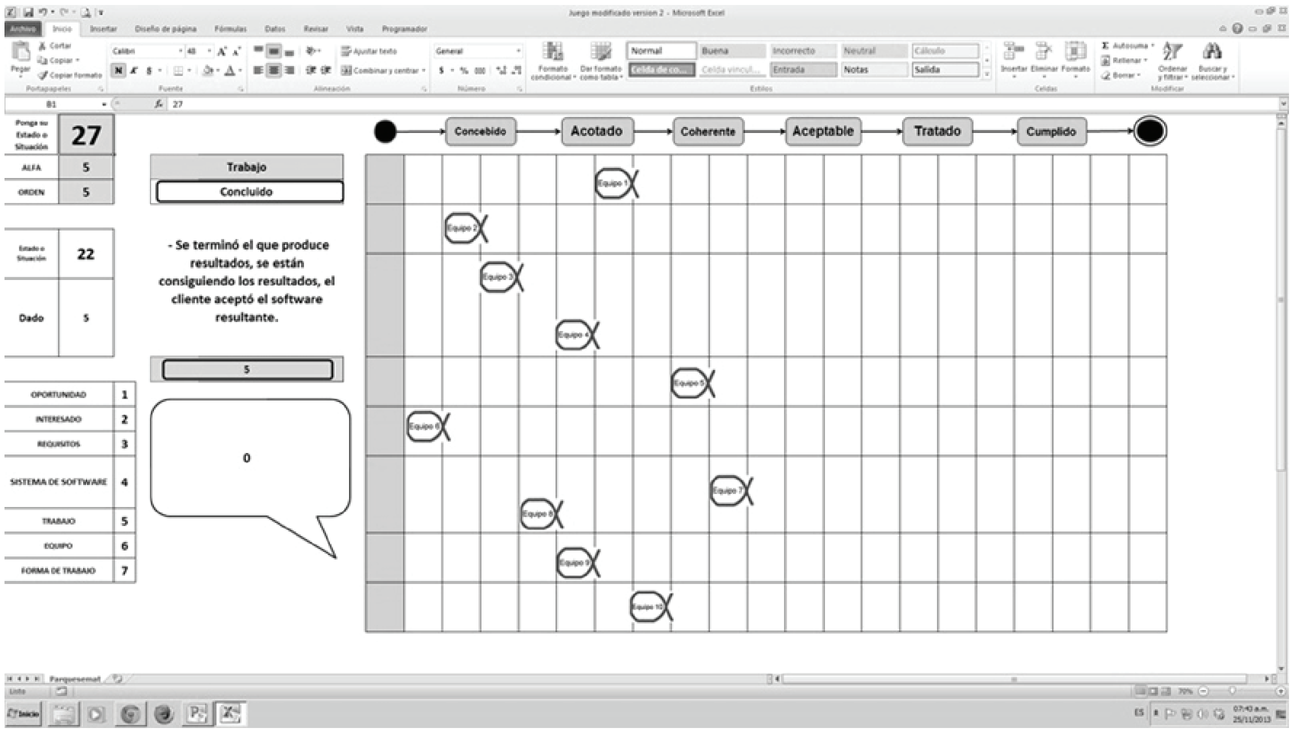
Courses vs. New methods. SEMAT kernel is not related to any particular method or practice. SEMAT kernel is a framework for describing any method or practice. So, there is no need to change the course contents with the emergence of new methods and practices.

Theory vs. Practice. Practical projects and games are strategies for promoting a close relationship between theory and practice. Particularly, some of the games selected for the course are based on real situations with the intention of encouraging decision making in simulated environments.


Sistema de Software: Retirado
Arquitectura Seleccionada
Sistema Demostrable
Sistema Utilizable
Sistema Listo
SISTEMA DE SOFTWARE OPERACIONAL
Definir avance
Evaluar situación de salud y progreso
Turno

Porcentaje a avanzar: 0 %
Porcentaje de avance: 0 %
EQUIPO : Verde

Figure 5. Image from the software system alpha game. Source: the authors.



Ponga su Estado o Situación	27
ALFA	5
ORDEN	5
Estado o Situación	22
Dado	5

Trabajo
Concluido

- Se terminó el que produce resultados, se están consiguiendo los resultados, el cliente aceptó el software resultante.

5

0

OPORTUNIDAD	1
INTERESADO	2
REQUISITOS	3
SISTEMA DE SOFTWARE	4
TRABAJO	5
EQUIPO	6
FORMA DE TRABAJO	7

Figure 6. Image from the requirements alpha game. Source: the authors.

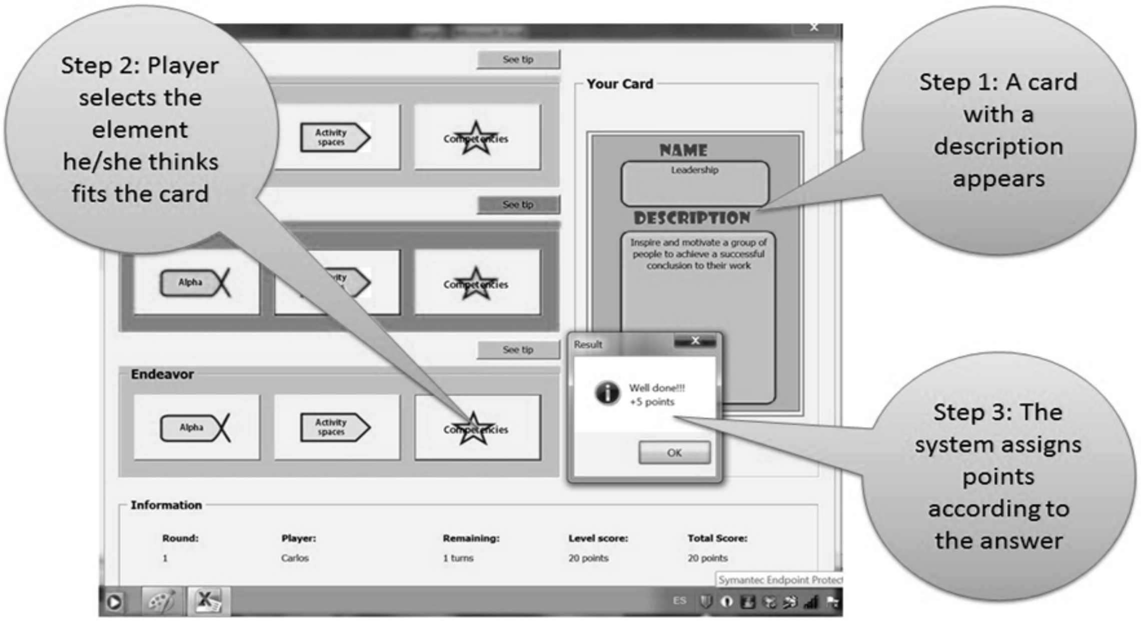

Step 2: Player selects the element he/she thinks fits the card
Step 1: A card with a description appears
Step 3: The system assigns points according to the answer
See tip
Your Card
Activity spaces
Competencies
NAME
Leadership
DESCRIPTION
Inspire and motivate a group of people to achieve a successful conclusion to their work
Alpha
Endeavor
Result
Well done!!! +5 points
OK
Information
Round: 1
Player: Carlos
Remaining: 1 turns
Level score: 20 points
Total Score: 20 points

Figure 7. Image from the SemCards game. Source: the authors.

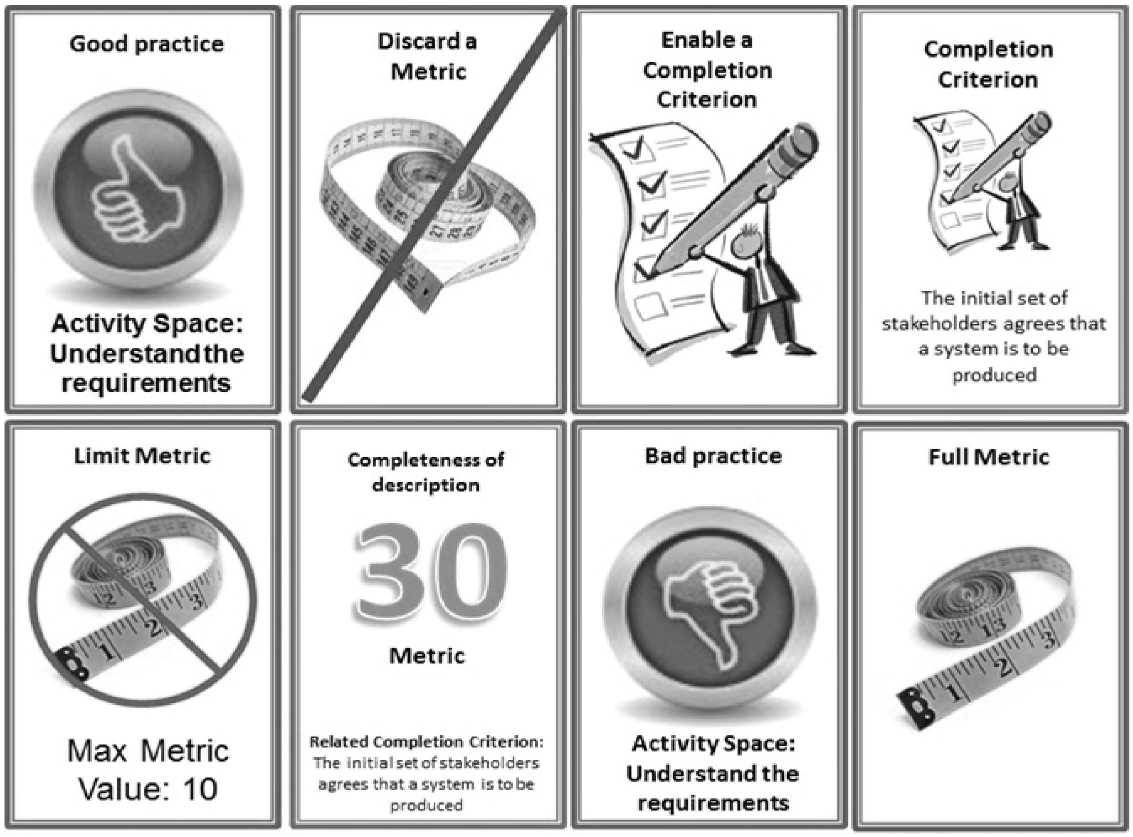

Good practice
Activity Space: Understand the requirements
Discard a Metric
Enable a Completion Criterion
Completion Criterion
The initial set of stakeholders agrees that a system is to be produced
Limit Metric
Max Metric Value: 10
Completeness of description
30
Metric
Related Completion Criterion: The initial set of stakeholders agrees that a system is to be produced
Bad practice
Activity Space: Understand the requirements
Full Metric

Figure 8. Image from the MetricC game. Source: the authors.

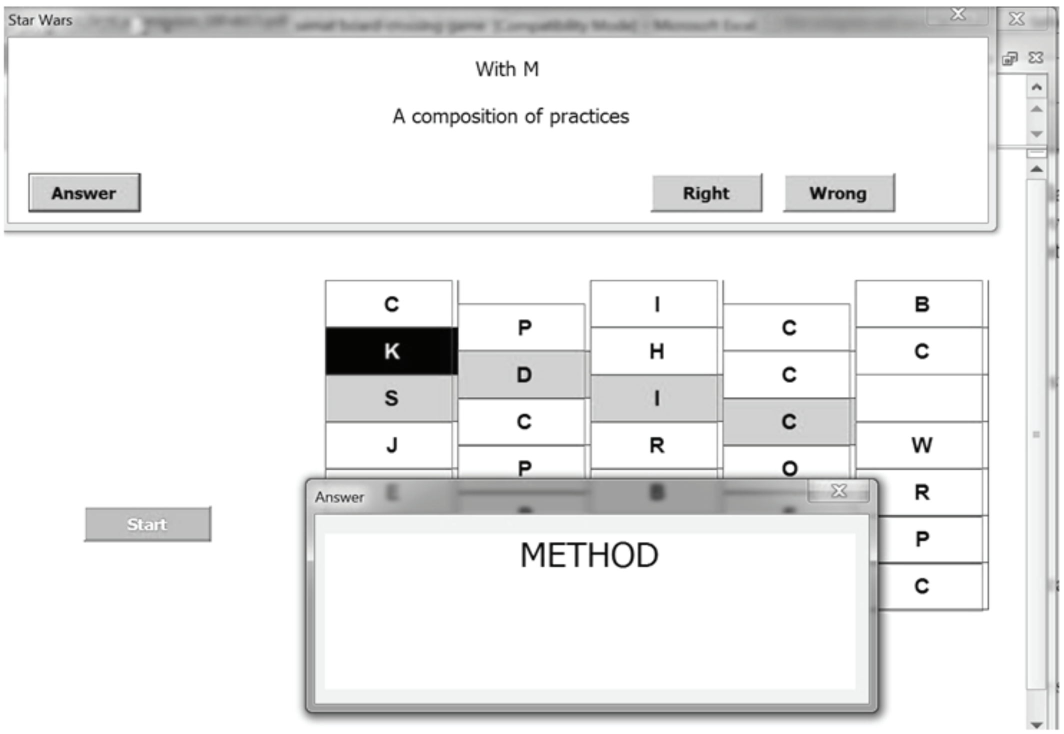


Figure 9. Image from the SEMAT board-crossing game. Source: the authors.

Software Engineering Theoretical Basis. By proposing a first course on SEMAT, we are promoting the dissemination of the SEMAT theoretical basis among students. The difference of this effort versus the previous ones is closely linked to the way in which we will try to teach software engineering. In fact, you can discover many software engineering courses related to specific methods and practices, but this is the first attempt to teach the theoretical basis underlying any methods or practices.

5. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a first course in software engineering methods and theory based on the work we are conducting in the Universidad Nacional de Colombia, Medellin Branch, related to the SEMAT initiative. The course is highly supported by the SEMAT community—for instance in the shape of lectures held by international guests—as a joint effort to keep a close contact among the SEMAT Latin American Chapter and the other regional Chapters. Also, the course thrives on the work we are conducting in games and other teaching strategies, because we are incorporating five SEMAT games created in the Universidad Nacional de Colombia.

We also discussed the way to deal with some common problems of the software engineering teaching, mainly related to the gap between software industry and academy, the continuous emergence of new software engineering methods, the scarce practice of the engineering newcomers and the lack of a software engineering theoretical basis

We propose as future work a complete assessment of the course results in terms of satisfaction surveys to students and professionals linked to the software industry. Also, the creation of new games to be included in the course contents can contribute to improve the SEMAT course.

REFERENCES

[1] Fairley R. Educational Issues in Software Engineering, en 1978 annual conference ACM/CSC-ER (1978, Washington D.C., USA). pp. 58–62.

[2] Mead, N. Software engineering education: How far we've come and how far we have to go The Journal of Systems and Software, 82, pp. 571–575, 2009.

[3] Ludewig, J. and Reißing, R. Teaching what they need instead of teaching what we like—the new software engineering curriculum at the University of Stuttgart, Information and Software Technology, 40, pp. 239–244, 1998.

[4] Amiri A., Banari M., and Yousefnezhad N. An Investigation of Undergraduate Software Engineering Curriculum: Iranian Universities Case Study, en International Conference on Computer Science & Education ICCSE (6th, 2011, Singapore, Singapore). pp. 638–644.

[5] Moreno, A., Sanchez-Segura, M., Medina-Dominguez, F., and Carvajal, L. Balancing software engineering education and industrial needs, The Journal of Systems and Software, 85, pp. 1607–1620, 2012.

[6] Jacobson, I., Ng, P. McMahon, P., Spence, I., and Lidman, S. The essence of software engineering: appying the Semat kernel, 1st edition, New Jersey, Addison Wesley, 2013.

[7] Zapata, C. M., Gelbukh, A., and Arango, F. Pre-conceptual Schema: A Conceptual-Graph-Like Knowledge Representation for Requirements Elicitation, Lecture Notes in Computer Science, 4293, pp. 17–27, 2006.

[8] Zapata C. M., Maturana G., and Castro L. Tutorial sobre la iniciativa SEMAT y el juego MetricC, en Congreso Colombiano de Computación (8th, 2013, Armenia, Colombia).

[9] Zapata C. M. and Montilva J. La esencia de la Ingeniería del Software: el núcleo SEMAT en la práctica, en Conferencia Latinoamericana en Informática (39ª, 2013, Naiguatá, Venezuela).