

# Enhancing Teaching & Learning of Software Engineering in an International Environment

Emanuel S. Grant, Washington I. Helps

**Abstract**— A fundamental artifact of any academic research is the data used as the basis of that research effort. A group of researchers, from institutions in multiple territories, has embarked on an ambitious research project that is initially aimed at enhancing the teaching of software engineering for four-year undergraduate programs. The research project details a set of workshops, for which the objective is the capture of data that will be the basis of the research effort. The first of these workshops was held in August of 2011, and is reported on in this paper. The workshop comprised software engineering educators and representatives from the information technology industry. The data collection task sought to identify a set of topics that are considered suitable for teaching software engineering, along with identification of the years and depth at which these topics should be taught. The use of technology in the curriculum design process is seen as an important step forward and this issue is also discussed.

**Index Terms**—Software engineering education, curriculum, course assessment, SDLC

## I. INTRODUCTION

FACULTY members at the University of North Dakota (UND), Department of Computer Science have been collaborating with colleagues at the Holy Angel University (HAU) in the Philippines and HELP University College (HELP) in Malaysia on teaching and research interests since the summer of 2007. Since then institutions in Ethiopia (Jimma University), Trinidad (University of the West Indies), India (International Institute of Information Technology, Hyderabad), and USA (Baylor University, Montclair State University, University of North Carolina Wilmington), have joined the collaboration.

The University of North Dakota, Office of Instructional Development is thanked for their partial sponsorship of the work done in preparing material for the workshop.

This group of software engineering faculty members, who are committed to the advancement of the teaching of software engineering (SE), has embarked on a project to develop and support creation and adaptation of learning materials and teaching strategies at the undergraduate level of higher

education. The goal of this project is the development of a teaching and learning online environment that will be open to all educators who are teaching in the field of software engineering. The underlining philosophy of the group's work is that the delivery of shared teaching material for courses that have been developed by the most knowledgeable in each topical area will advance the learning goals of such courses, and advance the adoption of teaching strategies that are in the students' best interest.

This works motivation comes from the group members experience in preparing and teaching courses and topics in the area of SE, across the four years of an undergraduate program in computer science and information technology. SE is emerging as one of the corner stone of computer science, yet the teaching of the subject is not standardized enough to facilitate a sharing of teaching resources across departments, nor institutions. This problem is exemplified by the diversity of topics, which may be taught in a software engineering course. If this problem can be solved then there would be an environment for the sharing and structuring of common software engineering curricula. The expected outcome of this proposed work is a framework for structuring undergraduate software engineering. For the purpose of this paper, Curricula include philosophy (goals), content (topics), approach (delivery) and assessment. Figure 1 embodies the definition of curriculum where the distinct segments encapsulate each major area. This framework will be a repository of essential software engineering teaching modules, assessment artifacts, course projects, and assignments.

A key goal of this research project is the identification of artifacts to be included, and the topical areas of SE that should be taught at the various levels of a four-year undergraduate degree. Towards this goal, the research group has launched a series of workshops in the territories of the participating institutions to capture relevant data. This report documents the input, activities, and output of the first such workshops that was held at HAU in August 2011.

## II. THE RESEARCH PROJECT

The research project has been partitioned into a Phase I and Phase II format, with Phase I being targeted at design of the framework for teaching undergraduate software engineering, including: core learning objectives, essential software engineering modules, along with detailed case studies, assessment artifacts, course projects and assignments. Phase

Manuscript received February20, 2012.

E. S. Grant is an Associate Professor with the Department of Computer Science, University of North Dakota, Grand Forks, North Dakota, ND 58202 USA, phone: 701.777.4133, fax: 701.777.333., email: [grante@aero.und.edu](mailto:grante@aero.und.edu).

W. I. Helps is a graduate student at the Department of Computer Science, University of North Dakota, Grand Forks, North Dakota, ND 58202 USA.

II of the project will address the design and implementation a repository for this information, and a series of training workshops for educators who intend to use the repository.

The workshops planned for Phase I of the project are the primary mechanism for the identification of these teaching and learning concepts and topics and the development of the framework. Through the broad-based approach of multiple workshops at the participating institutions, it is expected that the most comprehensive set of software engineering teaching and learning concepts and topics will be identified. At each workshop, the widest range of educators will be invited to participate. Each workshop will start with the broadest range of concepts and topics and the effort will be focused on the identification of a commonly agreed subset that captures all the concepts and topics that are deemed fundamental to the teaching of software engineering at the course and program levels. The projected timeframe for Phase I is three years, covering the period August 2011 – July 2014.

A first seminar was hosted at the Holy Angel University, Philippines July 2010. This seminar had in attendance ninety-five students and faculty members from twenty educational institutions in the central Philippines region, along with representatives from government agencies and business professionals. There were two main presentations, a hands-on tutorial, and a panel discussion from researchers of this project. The first presentation [1] examined the funding and research trend in SE in the USA, and looked at what impact the collaboration between Asian researchers and their United States counterparts may have on the research trends. The second presentation [2] looked at a novel approach to teaching object-oriented analysis and design as a topic of a SE course. The hands-on tutorial [3] took the participants through the fundamental steps of developing an online course. The panel discussion fielded a series of questions and answers from the areas of the two main presentations.

#### A. *The Repository*

The second phase is the Repository development phase of the research project. The Repository is intended to enhance and advance the teaching and learning of SE topics at the undergraduate level, by providing a collection of artifacts that captures the best practices in the field. SE educators will be able to access and use the Repository to develop course curriculum and syllabus, or enhance existing courses with high quality teaching and learning components. Educators will be able to select items for teaching and learning across the four years of an undergraduate program. The Repository will store items for curricula development (course topics, concepts, principles, etc.), syllabi development (recorded lectures, test/exams, assignments, term projects, term papers, course text, etc.), and course assessment material (student surveys/questionnaires, course evaluations, etc.).

Access to the Repository will be over the Internet, which will also provide the gateway for requesting/suggesting updates to the Repository (addition/deletion of items, and modification of existing items). The architectural platform for

the Repository will be Cloud, and associated computing technologies [4]. Cloud Computing, software as a service (SaaS) [4], and service-oriented architecture (SOA) [5] are growing phenomena in the business world. In contrast to traditional software systems, where organizations maintain technical staff and enough computer hardware to run their business applications effectively, SaaS and Cloud Computing are beginning to change this traditional way of managing information technology (IT). SaaS has an incentive to release new features as soon as they are completed to the users.

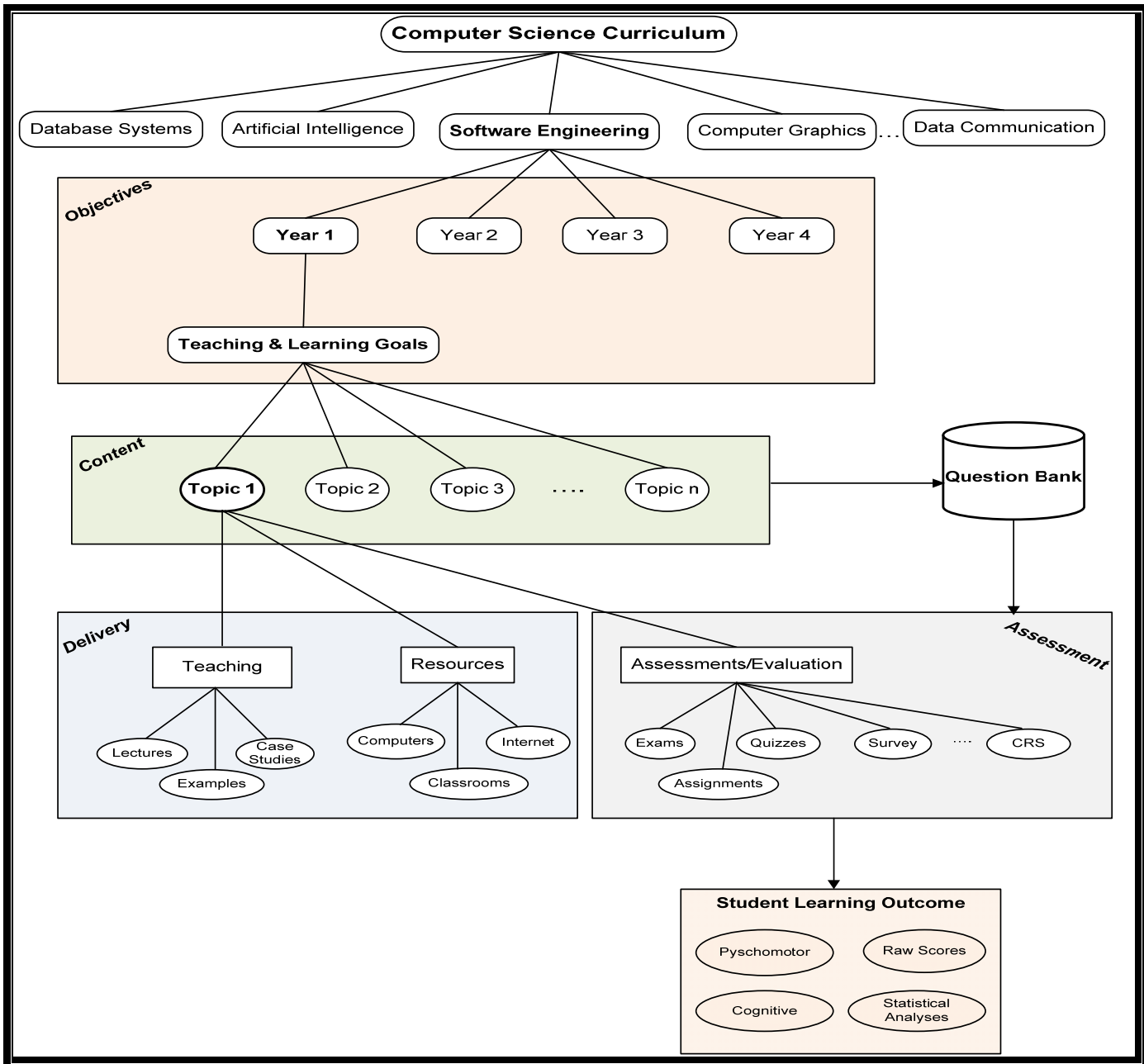
#### B. *Rationale*

Despite the advances in technology, there still seems to be limited utilization of such technology in the field of SE education. While there is a belief that technology could improve the educational productivity and help schools to teach more efficiently, evidence to support this belief is scarce [11]. The harnessing, utilization, and integration of technology would prove beneficial to the development, delivery, and assessment of education in a broad spectrum. The scarcity in human resources in the way of lecturers and experts in the related field has been a barrier to the effective delivery of standardized excellent curriculum in many territories. Sub-standard and limited physical resources such as communications equipment (broadband Internet, wired or wireless phone infrastructure) in varying degrees has also contributed to the educational gap and lack of coherency, which exists between developed and developing territories. Furthermore, within the borders of a developing territory, one may find that the expertise and available resources are only available in the urban centers, thus limiting accessibility and compounding macro-economic issues such as ‘brain drain’.

This gap can be expounded by examining the scenario in which a student with an undergraduate degree from a South Eastern nation is not viewed in the same way as a student from the United States, who has a similar undergraduate degree in the same discipline. It is for this reason that many colleges require a Graduate Comprehensive examination for students wishing to matriculate into a graduate program, ensuring that all students in their graduate program have successfully mastered the undergraduate level programs regardless of where such programs were taken. This incongruent view is not necessarily a matter of culture, but one for which there is some merit because of the fact that as things stand currently, two courses having the same name and taught in two different territories does not mean that the content or delivery is the same.

An abstract view of the research project framework is depicted in Figure 1 in which a hierarchical outline is given. In this work, the SE curriculum in the field of Computer Science will be used as the working prototype. Thus for a given four year undergraduate degree program, a system derived from this framework will be able to for each year:

- Present all the teaching and learning goals
- Provide access to all relevant content by set of topics



**Figure 1** Hierarchical Map of Research Curriculum

- For each topic, show the recommended delivery (teaching) methods and required resources. For topics, which are not selected, alternatives may be suggested, but regardless of whichever topics constitutes the content, the coverage metric (a measure of how much of the curriculum is covered or selected) will be given.
- Recommend relevant assessment methods and provide automated assessments, which are mapped directly or indirectly to the content.
- Provide analysis of student learning outcome, which allows for a comparison between performance and expectations.

A focus of this work will be the curriculum development component. Another key component is assessment, which will encapsulate various assessment methods which can be viewed as an intermediary between the curriculum and the

student learning outcome, essentially allowing for a comparison between performance and expectations (purposes/goals). A foundational principle of this system is the notion of hierarchy or better yet, granularity. The levels of abstraction are very important and so the artifacts of the system can be viewed from top-down perspective.

Questions this Research will answer include:

1. How are course curricula designed?
2. Can they be modeled in the top-down tree-like structure?
3. How can the content of a curriculum in terms of topics be prioritized and tagged with weights in order for their relevance and importance to be explicitly noted and tracked.
4. How can the quality of a curriculum be maintained even when not all topics are covered (selected) or not all the resources are available? That are, how are possible trade-offs

handled?

5. Is it possible or feasible to have a concrete content to assessment mapping?

6. How are courses assessed, and what is the best assessment method/s for a given course?

7. What percentage course coverage (in terms of assessment) is necessary to qualify a student as having mastered a course or to make any other determination about a student's knowledge or aptitude?

### C. *Integration of Technology in Curriculum Development*

This framework will seek to make it easy for Software Engineering curricula to be developed while not taking away or eroding the quality of the curricula itself or the standards of the curricula regardless of where in the world it is being developed. There is the need for a framework in which the development of a curriculum can seamlessly accommodate cultural differences and diversities without negatively affecting the standard and quality of said curriculum. A curriculum developed and utilized in Thailand may work well in a Thailand setting, but may not work well in the United States for instance. So even though certain features may be different such as the delivery, resources and so on, the quality, standard and overall effect must be similar so that regardless of which ever territory the curriculum is based, its uniformity can be guaranteed.

Richard Gluga et al. have developed an architecture and a tool called CUSP (Course and Unit of Study Portal) [12] that serves a curriculum information tracking system which facilitates the systematic tracking of skill and competence level progression in a Computer Science context. The work in [12] touches on some issues that may be important to this research and some of the methods may certainly be useful at various stages of this effort. The issue of effectively modeling curriculum skills, mapping them to assessment tasks across subjects of a degree, and measuring the progression in learner competence level is seen as an unsolved problem for the most part. A system called ProGoSs is also utilized in [12], which is under active development. ProGoSs engrains Bloom's Taxonomy that is particularly important since Bloom plays a key role in defining curricula like the current ACM/IEEE-CS curriculum guidelines.

### D. *The Benefits*

This research effort has many potential benefits and systematically modeling a curriculum in the way proposed will allow for some benefits listed below:

1. Meaningful and even autonomous assessment of students. Because of the scarce resources in developing countries for example, educators do not have the time to give individual attention to all students in terms of post assessment, feedback, and overall progression evaluation.
2. Make informed decisions about curriculum. Feedback from system can be used to guide improvements, make changes or help educators zoom in (pay closer

attention) to problematic or weak areas.

3. Standardization of the delivery of curriculum. Thus maximizing on the expert knowledge in the particular field in which the system is utilized. For instance, a student doing a BSc in CS with a SE major in Jamaica (a developing country) and being taught by faculty with MSc degrees should have a similar learning outcome as a student in the USA (a developed country) doing a similar program of study and taught by faculty with PhDs.
4. Educators will be able to use this system to better understand students and make predictions about student performance.
5. Automated generation of assessments (test, exams etc). Educators need only specify what topics they need the students tested on and a randomized, heuristics based set of question will be produced. Since the question, Bank will be mapped to the Curriculum.
6. Automated statistical reports/analysis, etc.

## III. THE FIRST WORKSHOP

The first data collection workshop of August 2011 lasted the entire day, with thirty participants. The morning plenary session was used to present the structure of the workshop, expected goals/outcomes for the workshop; and describes the material that was used in the workshop. The thirty participants that comprised SE educators and recent graduates of the HAU IT program were divided into four panel groups for the afternoon breakout sessions. The two breakout sessions were followed by a presentation and wrap-up plenary session.

### A. *Workshop Preparation*

Prior to the date of the workshop a sub-group of the research project convened meetings at the HAU campus to plan the workshop. This planning involved the identification of suitable material to be used by the participants, design of instruments for collecting the data, plan the moderation of the breakout panel sessions, assign responsibilities to the participating researchers, and finalize all administrative functions.

The workshop preparatory team first tackled the task of identification and selection of material to be used in the workshop, as a guide for the work of identifying an appropriate set of topics for teaching SE across the four years of an undergraduate program. Some of the material examined included Effective Grading by B. E. Walvoord and V. J. Anderson [6], The Course Portfolio by P. Hutchings (ed.) [7], and The Teaching Portfolio by P. Seldin [8].

It was finally decided that the Institute of Electrical and Electronic Engineering Computer Society (IEEE-CS) and Association of Computing Machinery (ACM) Software Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (IEEE-CS/ACM SE 2004) [9] would be the main source of input data for the workshop. This document is the second version of the

original document produced in 1991, and is currently under review for the release of a more updated version. The IEEE-CS/ACM SE 2004 primary purpose is to provide guidance to academic institutions about what should constitute an undergraduate SE education.

The IEEE-CS/ACM SE 2004 defines nine subject areas, which are the Software Engineering Education Knowledge (SEEK) areas [9]. The SEEK areas are: Computing Essentials, Mathematical & Engineering Fundamentals, Software Modeling & Analysis, Software Design, Software Verification and Validation, Software Evolution, Software Process, Software Quality, and Software Management. Each of these areas is further decomposed into a set of topics, and supplemented with recommended hours of teaching, Blooms Taxonomy classification (k- Knowledge, c- Comprehension, a – Application), and measure of relevance (E – Essential, D – Desired, and O – Optional).

**B. Workshop Activities**

The main activities of the workshop were the breakout panel sessions and the wrap-up plenary session. For the breakout, panel sessions the participants were divided into four groups that conducted discussions in the following SEEK areas:

- Computing Essentials,
- Software Modeling & Analysis and Design,
- Software Verification & Validation, and Evolution,
- Professional Practice and Software Management.

Each Panel had a Moderator, whose tasks were the following:

- Guide the discussions through the list of topics in a timely manner,
- Record the decisions and votes taken by the panel,
- Seek clarification and issues that are not resolved,
- Participate in the discussions.

Each panel group discussed the topics listed under the respective heading, from the SEEK areas. The groups answered the following questions:

- Is this topic relevant to teaching the fundamentals of SE?
- In which year(s) of the program should it be taught?
- How many hours should be taught in each year?
- What is the rationale for the selection of this topic?

In the wrap-up plenary session, reports from the Moderators were presented, so that each participant had knowledge of how the discussions went in the other groups and the decisions made.

All the data will be posted on a, to be established workshop website, for all to review, use, and submit comments. The website will host information on conferences at which workshop reports are presented.

**C. Workshop Input/Output**

The single input used for the breakout panel sessions was sections from the IEEE-CS/ACM SE 2004 document. Twenty-two pages from the one hundred and thirty five-page document were distributed to each of the participants, and the

last section of the workshop morning session was spent reviewing this document. The twenty-two pages comprised introductory information on the IEEE-CS/ACM SE 2004 manuscript, and description of the SEEK, along with tables of the SEEK areas, with the supplementary information. Samples of the SEEK areas, as used in the workshop, are presented in tables 1 and 2.

TABLE I. SAMPLE SEEK KNOWLEDGE AREA

KA/KU	Title	hrs
<b>CMP</b>	<b>Computing Essentials</b>	<b>172</b>
CMP.cf	Computer Science foundations	140
CMP.ct	Construction technologies	20
CMP.tl	Construction tools	4
CMP.fm	Formal construction methods	8

The first column of Table I lists the Knowledge Area/Knowledge Unit (KA/KU), which is coded for easy reference in the documentation. The Title column lists the topic area, and the third column (hrs) specifies the recommended number of lecture hours for the topic.

TABLE II. SAMPLE SEEK KNOWLEDGE AREA: DETAIL

Reference		kc,a	E,D,O	hrs	Related
<b>CMP</b>	<b>Computing Essentials</b>			<b>172</b>	
CMP.cf	<i>Computer Science foundations</i>			140	
CMP.cf.1	Programming Fundamentals (control & data, typing, recursion)	a	E		
CMP.cf.2	Algorithms, Data Structures/Representation (static & dynamic)	a	E		CMP.ct.1, CMPfm.5, MAA.cc.1
CMP.cf.3	Problem solving techniques	a	E		CMP.cf.1
...	...	...	...		...
CMP.cf.13	Semantics of programming languages		D		

Table II presents the detail topics of the Computer Science Foundation sub-section of the SEEK Computing Essentials area. Column 1 is the coded identifier of the topic. Column 2 gives the topic title. Column 3 specifies the Bloom’s classification. Column 4 states the topic’s relevance. Column 5 lists the recommended lecture hours, and column 6 notes the related topics.

The panel groups went through the detail SEEK tables, deliberating over each topic and adjusted the entries in columns 3, 4, 5, and 6. One panel group made changes to the topic description of column 2, for a few entries. The panel groups also added an additional column to the original SEEK topic tables to identify the year(s) in which the topic should be taught. The panel participants deliberations were to identify a most appropriate (based on the group’s experience) set of topics, using the IEEE-CS/ACM SE 2004 SEEK areas format. The specific instruction asked them to define the perfect SE curriculum, by topics.

Table III was produced by one of the panel groups as the result of their deliberations on Professional Practice and Software Management. It is to be noted that they changed the column hrs to Depth, and assigned percentage of teaching time, instead of numeric values of hours. This was the decision taken in the first plenary session, as different institutions ascribe different quantities of lecture hours to each topic. Overall, the four panel groups were at a high



percentage of agreement with the original SEEK topics. The areas of greatest divergence were with respect to the Bloom's taxonomy classification and percentage of teaching time for each topic.

TABLE III. SAMPLE WORKSHOP OUTPUT TABLE

Professional Practice			
TOPIC	YEARS	DEPTH	RATIONALE
PRF.psy	1 to 4	15%	# 5 & #6 be change from K to C- >"DEALING" is not merely KNOWING. Dealing needs to provide course of actions to certain issues e.g. forecasting, predicting, identifying, presenting alternative solutions. All items are ESSENTIAL
PRF.com	1 to 4	40%	All items are ESSENTIAL
PRF.pr	1 & 4	45%	All items are ESSENTIAL
Software Management			
TOPIC	YEARS	DEPTH	RATIONALE
MGT.con	2 to 4	10%	All items are ESSENTIAL
MGT.pp	2 to 4	31%	All items are ESSENTIAL
MGT.per	2 to 4	11%	All items are ESSENTIAL
MGT.clt	2 to 4	21%	All items are ESSENTIAL #5 & #6 be change from O to E and classified as K->to achieve quality performance/output would require effective supervision/management
MGT.cm	2 to 4	27%	All items are ESSENTIAL #7 be change from D to E and classified as A->Security is an essential part of software configuration management. Topic #7 must be renamed as Distribution, Back-up, and Security. #4 be change from C to K->BUILDS deals more on conceptual/theoretical rather than actual application in the classroom set-up

Three of the panel groups included a recent graduate of HAU's IT program, who is currently working in an IT related field. The inclusion of such participants facilitated the assessment of the appropriateness of each topic from the "students' point of view. These participants were specifically charged with commenting on how useful they found the topics in their current profession (if it was taught in their program), at what point in their program they thought it was most suitably taught, and was the delivery (teaching) of the topic done in a manner that made them interested.

IV. CONCLUSION

The workshop conducted at the HAU campus in August 2011 by an international team of researchers to identify an appropriate set of topics for teaching software engineering across the four years of an undergraduate program was successful.

During the panel groups' presentation at the closing plenary session, it was noted that though the groups conducted their work separately and privately, the deliberations were conducted in a similar manner. The participants felt the exercise was productive and necessary in advancing the teaching of SE in their respective institutions. A number of the participants also expressed a desire to incorporate some of the ideas and topics introduced at the workshop, in their

current SE courses and program.

The research group will commence analysis of the data that was gathered for the specific purpose of refining the workshop structure and process, towards implementing future similar workshops to validate the findings from this first one. It is planned that after the workshops have been completed an annual international conference will evolve out of this effort. It is planned for this conference to be hosted in the Philippines between HAU and other major participating institutions in the Philippines. The research group will continue its out-reach effort to recruit SE researchers and educators from institutions in other territories to participate in the workshops and research effort.

ACKNOWLEDGMENT

The research expresses thanks to the workshop participants, especially the three new graduates (Karla Sinohin, Ivy Serrano, and Karen Castro), who so ably worked with their former professors. The research group also expresses it thanks to the staff of HAU Graduate School who provided administrative and secretarial support to the workshop.

REFERENCES

- [1] Grant, E. S. (2010). Trends in Software Engineering Research, International Seminar on Software Engineering Development and its Application in Education, Holy Angel University, Philippines.
- [2] Sien, V. Y. (2010). Guidelines for teaching object-oriented analysis and design, International seminar on software engineering development and its application in education, Holy Angel University, Philippines..
- [3] Salta, M. T. (2010) Online Courseware Development, International Seminar on Software Engineering Development and its Application in Education, Holy Angel University, Philippines.
- [4] Mohammad A. F., & Grant E (2010). Cloud Computing, SaaS, and SOA 3.0: A New Frontier Proceedings of the International Conference on Cloud Computing and Virtualization 2010, Singapore
- [5] Mohammad A. F., & Grant E (2010). Cloud Computing, SaaS, and SOA 3.0: A New Frontier Proceedings of the International Conference on Cloud Computing and Virtualization 2010, Singapore.
- [6] B. E. Walvoord and V. J. Anderson, Effective Grading: A Tool for Learning and Assessment. Jossey-Bass, A Wiley Company, CA: 1989.
- [7] P. Hutchings (ed.). The Course Portfolio: How Faculty can Examine their Teaching to Advance Practice and Improve Student Learning. The American Association of Higher Education (AAHE). Washington DC. 1998.
- [8] P. Seldin. The Teaching Portfolio: A Practical Guide to Improve Performance and Promotion and Tenure Decisions. Anker Publishing Company, Inc. Bolton MA. 2004.
- [9] Joint Task Force on Computing Curricula. Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE Computer Society and ACM. 2004.
- [10] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, Merlin C. Wittrock. A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon. 2000.
- [11] B. Hokanson, S. Hooper. Integrating Technology in Classrooms: We Have Met the Enemy and He is Us. Association for Educational Communications and Technology, 27th, Chicago, IL, October 19-23, 2004.
- [12] R. Gluga, J. Kay, T. Lever, R. Lister, and T. Lever. An architecture for systematic tracking of skill and competence level progression in computer science. In Computer Science Education: Innovation and Technology CSEIT 2011, December 12, 2011.