Towards an Online Collaborative Software Engineering Teaching Environment

Emanuel S. Grant

***Abstract*—** The profession of software engineering is one of the faster growing in the world; in the United States, it is projected that there will be over two million unfilled computer technology positions by the year 2020. Educational institutions are working to address this increasing demand for graduates to fill the personnel vacuum. An approach being pursued by a group of researchers is to develop an open-source Cloud-based repository of teaching artifacts. The repository will be a relational database of searchable case-based material that are tied to related program/course outcomes/topics and structurally attached to assessment material. Other projects have offered similar products and services, but this project focuses on sustainability by designing a dynamic content repository with broad participation.

***Keywords*—**Software engineering, curriculum, pedagogy.

I. Introduction

The pervasiveness of technology in society demands a workforce that is capable of addressing the increasing demand for products and services that are worldwide applicable. The Internet World Stat website shows there has been an 832.5% increase in worldwide Internet usage between 2000 and 2015, with a corresponding increase of 1,319% increase in Asia, over the same period. A description for some Asia countries is presented in Table 1.

TABLE I: ASIAN COUNTRIES INTERNET USAGE INCREASE

Country	Population	Users 2000	Users 2015	% Change
China	1,361,512,535	22,500,000	674,000,000	2,895.55
Hong Kong	7,141,106	2,283,000	5,751,370	151.92
India	1,251,695,584	5,000,000	375,000,000	7,400.00
Japan	126,919,659	47,080,000	114,963,827	144.19
South Korea	49,115,196	19,040,000	45,314,248	138.00
Malaysia	30,513,848	3,700,000	20,596,847	456.67
Philippines	109,615,913	2,000,000	47,134,843	2,256.74
Singapore	5,674,472	1,200,000	4,653,067	287.76
Taiwan	23,415,126	6,260,000	19,666,364	214.16
Thailand	67,,976,405	2,300,000	38,000,000	1,552.17

Table 1 illustrates the growing population of Internet usage, which is an indication of the growing application of computing technology. This implies a growing need for educated software developers to design, implement, and maintain these systems. The Swiss nonprofit foundation, World Economic Forum listed 14 technology predictions for the year 2020 and six of them were directly computer technology related (job learning, Internet of everywhere, data-driven healthcare, Internet of things, tele-communication, humanized Internet). A Computerworld article by Patrick Thibodeau [1] presents a breakdown of jobs figure in the computer technology industry for the year 2020 that is derived from official United States government figures. These figures predict an average growth of 22% in the computer technology fields, with the highest percentage growth of 32 in the software developers'/systems software area.

These statistics and their implications are justification for the work presented in this paper. The work seeks to address the need for educated personnel in the information technology, specifically in the sub-discipline of software engineering (SE). The research effort seeks to develop a repository of teaching artifacts for SE education. The repository will implement an ontological relationship across a heterogeneous collection of teaching material that facilitates searching and selection of a subset of the repository content. This selection will be driven on a set of specified program/course outcomes for a four-year undergraduate-level of education in SE. The key artifact of the repository will be a set of case examples for SE that is ontologically related to the desired outcomes, topics, lectures, and assessment instruments.

The next section of this paper provides the background and related works to the research project and that is followed by the section on the methodology of the research. A section on current state of the research precedes the conclusion and future work.

II. Background

A. Problem Statement

Software engineering is viewed as one of the corner stones 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 courses. If this problem can be solved by the establishment of a framework for developing software engineering teaching material (lectures, examinations, assignments, and course projects), teaching assessment process, and delivery platforms then there would be an environment for the sharing and structuring of common software engineering curricula.

In academia, educators in software engineering have to address a number of questions in selecting an appropriate software development methodology for teaching at the

Emanuel S. Grant, University of North Dakota, USA, Email: emanuel.

graduate and undergraduate levels. Some of these questions are: (1) which of the available methodologies includes a suitable set of processes, techniques and notations for teaching and demonstrating the fundamental concepts of software development? And (2) what processes, techniques, and notations are necessary to teach and demonstrate the fundamental concepts of software development? This situation is aggravated by the numerous books that purports to provide answers to the preceding questions – A search for "software development" on four popular publishers websites produced the following listing: Addison-Wesley (http://catalogs.mhhe.com/mhhe/home.do) 106 books, CRC Press (www.crcpress.com) 150 books, Prentice Hall (http://vig.prenhall.com/) 70 books, and O'Reilly (http://www.oreilly.com/) 261 books.

With such a large number of choices, educators may make selection of a methodology based on: (1) their experience with a particular methodology, (2) knowledge of what others have used, (3), familiarity with a researcher's or authors' work, (4) recommendation of other educators, and (5) research of the available methodologies. With the exception of (5), which may be carried out with varying intent, the others do not address the key questions of the previous paragraph. The availability of a description of what constitutes an appropriate set of processes, techniques, and notations for software development will be a significant aid to educators in teaching the essential and advance concepts, as well as selecting appropriate teaching material.

The teaching research of this proposal is similar to that being conducted by the Software Engineering Discipline Commons - SEDC (http://sec.cs.siue.edu/) at the Southern Illinois University at Edwardsville. Wherein SEDC limits itself to institutions within a geographical region, this research is intended for international collaboration. A further difference between the two research intents is the SEDC delivers a static collection of teaching and scholar material, whereas this proposed research will deliver a dynamic set of artifacts, which may be updated by any registered user of the repository.

B. Expected Outcomes

The expected outcome of this proposed work is a framework for structuring software engineering curricula. This framework will be a repository of essential software engineering teaching modules, assessment artifacts, course projects, and assignments. The repository will be available online for open use, and the completed research work and progress will be presented at conferences at which it has been accepted. Workshops on the proposed work will be conducted by the PIs in the three territories.

The proposed software engineering teaching research will result in an integrated environment that leverages an identified set of best practices, with respect to software engineering teaching techniques, processes, and material towards the definition of collaborative teaching methodology. The research will cover issues from the point of requirements elicitation through system integration testing. The phases of software engineering teaching that are specified in this research are: (1) requirements engineering (elicitation, analysis, and specification), (2) requirement model design (architectural, component, and interaction), (3) object-oriented model analysis and design (static and dynamic), (4) system verification and validation (formal and informal) (5) implementation (model to code transformation, and code testing), (6) system maintenance, and (7) management (quality management, process improvement, and configuration management).

The repository will be an electronic searchable collection of teaching material, which may be assembled to fit the requirements of a particular curriculum for teaching software engineering. An ontological of the repository artifacts will facilitate meaningful selection of course material, in building a curriculum for a semester course in software engineering. The goal is to make this repository available, via registration, over the Internet. Practitioners of software engineering teaching may choose to collaborate on teaching one such course, or chose to teach it with one instructor. In the case of this proposed work, there will be collaborative teaching of the course between the three campuses.

C. Related Research

The Software Engineering Disciplinary Commons (SEDC) is a project that was funded by the National Science Foundation (NSF), and the program has two stated objectives. The first is the documentation and sharing of knowledge about student learning on courses in software engineering in four-year degree granting institutions within a single geographic region. The second objective is to improve the quality of teaching in software engineering (and generally in computer science) by establishing practices for the scholarship of teaching by making it public, peer-reviewed, and amenable for future use and development by other software engineering educators.

The mechanism for achieving the project goals is via the shared production of course portfolios. This mechanism will be instantiated through a series of monthly seminars involving a number of faculty members at baccalaureate-degree granting universities in the Mid-America / Mid-West region of the United States. Participants will meet once per month over an academic year. At these meetings, participants will learn how to construct a course portfolio, skills of classroom assessment and peer reviews, and will critically evaluate one another's work-in-progress.

The course portfolio, is a well-known method for advancing teaching practice and improving student learning. It is a document that reflects the content of a single course from its conception to outcomes. Course portfolios typically include a course's learning objectives, its contents and structure, a rationale for how this course design meets its objectives, and the course's role in a larger degree program. Importantly, the portfolio also includes evaluations of student work throughout the term, indicating the extent to which students are meeting course objectives and the type and quantity of feedback they are receiving. Each participant in the project will construct a

course portfolio for a course that they teach that is on the path for a baccalaureate degree in a Computer Science program.

The power of the portfolio approach is multiplied when there are several examples available for a single disciplinary aspect: The Commons would act as a public and peer-reviewed portfolio repository and archive, charting and calibrating excellence over time. Further, by linking educators between community colleges and universities, a regional network of educators within a single discipline will be strengthened for future collaboration and knowledge-sharing.

The work of this proposal encompasses that of the SEDC, and adds three more layers; namely – (1) international participation, (2) collaborative distributed teaching, and (3) relational database retrieval and update of teaching material. One of the PI is a participant in the SEDC project, and he will be incorporating the learning from that project into this proposal work.

III. Methodology

The international collaborative teaching paradigm will consist of a repository of teaching material that represents the identified commonalities and variabilities that have been identified as essential for the teaching of software engineering. The topography of this repository will be an ontologically structured set of teaching and assessment artifacts that is based on a set of desired program/course outcomes. A key element of this ontologically structured repository of artifacts will be the use of cases around which the outcomes will be realized. Figure 1 illustrates the ontological relationship that will be implemented in the repository between the teaching and assessment artifacts. The supporting content at the bottom of the ontology map includes the case examples that are now identified as a fundamental artifact of the repository.

A goal of the research work is providing users with the facility to develop program/course curriculum within a technology framework. Figure 4 illustrates the hierarchical organization of the components of a curriculum. The body of knowledge that is deemed as appropriate for an undergraduate program in software engineering is designated in SEEK [2]. SEEK is organized hierarchically into three levels. The highest level is the education Knowledge Area, which represents a particular sub-discipline of software engineering that is generally recognized as a significant part of SE knowledge that an undergraduate should know. The second level is the units where each knowledge area is broken down into smaller divisions or modules.

Each unit is then subdivided to form the lowest level, which is a set of topics. The research introduces case examples that are linked to the topics; the case examples provide the connection between theory and practice.

Figure 5 provides an overview of the methodology. Phase 1 details the data capture for the repository. This builds on the work of Grant, et. al. in [3] and [4] in order to outline a path to the realization of the repository. Phase 2 implements the repository development and population, given the input from Phase 1. Finally, Phase 3 illustrates utilization of the repository for the delivery of the content and student assessments as well as automated feedback to the repository.

A. Benefits

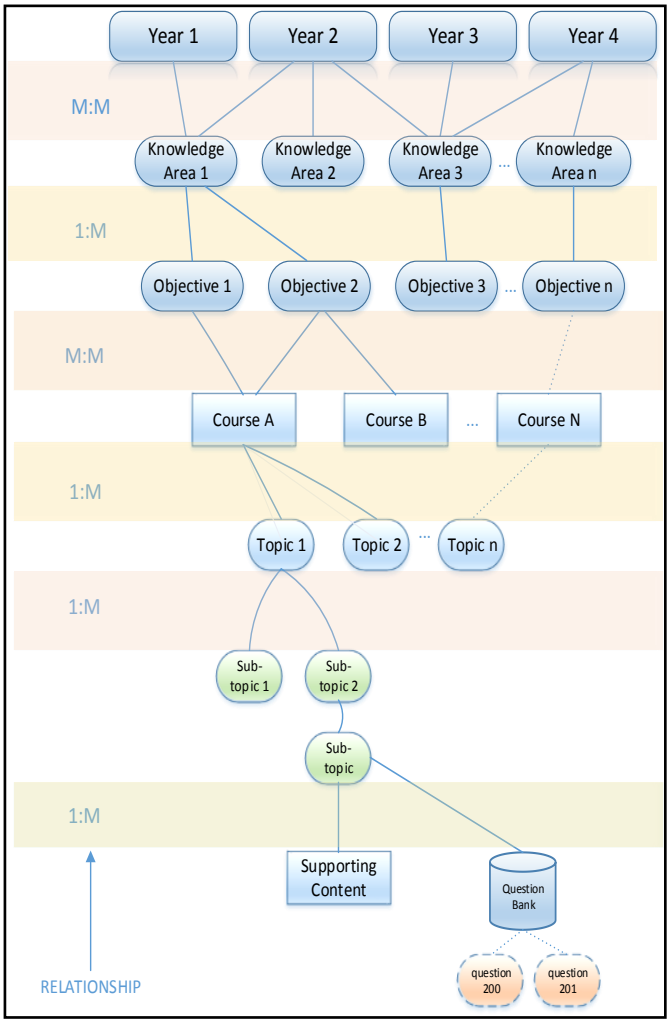

Fig. 1. Repository Topography

The community of software engineering pedagogy at the graduate and undergraduate levels is the target group of this proposed work. The availability of the project repository will be useful to tow groups in this community. The two groups are new and intermediate teachers of software engineering and the experience teachers of software engineering. The new and intermediate group will benefit by being able to access a researched, searchable, and retrievable collection of software engineering artifacts to either start or improve their teaching of the subject. The group of experienced teachers will be able to add their teaching experience to the collection of material, and thus share their knowledge with an international community of educators.

There are many issues faced by computer science educators, and in particular software engineering educators. One of these issues is the selection of software engineering teaching material, e.g. course texts. Another is the identification of an essential set of topics and concepts for the teaching of software engineering. The presence of a common approach to teaching the essentials of software engineering is also one of

the major issues faced in this discipline. A final issued of concern is that of being able to conduct collaborative teaching that leverages the best of multiple contributors. These are the main issues that will be addressed in this proposed work on teaching software engineering. It is expected that by addressing these issues a school of material and methodology will become available that will advance the teaching of software engineering that is evolvable, dynamic, and stay current.

The proposed project schedule identifies a twelve-month project lifetime, and a twelve-month post project evaluation period. The twelve-month post project evaluation covers the collaborative teaching of the initially designed software engineering curriculum across the three campuses that are involved in the project. The three semesters overlap and spans the eleven-month period, inclusive of three months of post teaching evaluation.

IV. RESULTS

The group has conducted a number of workshops, as, as described in Phase I of the project. These workshops have produced vital information that will be used as input to Phase II activities, of developing the repository. In the first workshop, the earlier version of the ACM/IEEE-CS document [5] was used as a seed document to the activities of determining commonalities and variabilities of SE education.

TABLE II: SAMPLE SEEK KNOWLEDGE AREA

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

TABLE III:SAMPLE SEEK KNOWLEDGE AREA: DETAIL

Reference		kc,a	E,D,O	hrs	Related
CMP	**Computing Essentials**			**172**	
CMP.cf	*Computer Science foundations*			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		

The single input used for the breakout panel sessions was sections from the earlier version of [2]. The section used comprised introductory information on the IEEE-CS/ACM description of SEEK, along with tables of the SEEK areas, with the supplementary information. Samples of the SEEK areas used in the workshop, are presented in tables 2 and 3. The first column of Table 2 lists the Knowledge Area/Knowledge Unit (KA/KU), coded for reference. The Title column lists the topic area, and column three (hrs) the recommended number of lecture hours for the topic.

Table 3 presents the detail topics of the Computer Science Foundation section of the SEEK Computing Essentials area. Column 1 is coded identifier of the topic. Column 2 is the topic title, column 3 the Bloom's classification, column 4 the topic's relevance, column 5 the lecture hours, and column 6 notes the related topics.

V. CONCLUSION

The work of this paper is an amalgamation of multiple research efforts that are geared toward the development of a repository of software engineering teaching and assessment artifacts. This report draws on research effort from [2, 5]. The project has collect data from a series of workshop that will be the input to Phase 2 of the project. This phase will result in the design, development, and population of the repository. A series of workshops will follow for the purpose of training and use of the repository. The ultimate goal is for sustainability through evolution of the repository.

REFERENCES

[1] P. Thibodeau, "IT jobs will grow 22% through 2020, says U.S.," Computerworld, [Online].

[2] Computer Science Curricula 2013, Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, the Joint Task Force on Computing Curricula, ACM and IEEE-CS, New Jersey, Dec. 2013.

[3] Grant, E. Online Collaborative Teaching of Software Engineering in an International Multi-Campus Environment. First HAU International Workshop on Enhancing Teaching & Learning of Software Engineering, Holy Angel University Angeles City, Philippines August 12, 2011.

[4] Grant, E., Stokke, T., Helps, W., Esquivel, J., Concepcion, W. Enhancing Teaching & Learning of Software Engineering in an International Environment. In Proceedings of the 2nd Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2011), pages 36-41, 2011.

[5] P. Bourque and R. E. Fairly, Guide to the Software Engineering Body of Knowledge (SWEBOK□): Version 3.0. Ed. IEEE Computer Society, New Jersey, USA, Jan. 2014.