Case Method for Computing Education (CMCE)

A Strategy for Teaching Software Engineering

Emanuel S. Grant
Department of Computer Science
University of North Dakota
North Dakota, USA
grante@aero.und.edu

***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 aimed at enhancing the teaching of software engineering in four-year undergraduate programs. The research project details a set of objective, with the first being the capture of data that will be the basis of the research effort. The first of these workshops was held in August 2011. The data capture activity involves work with 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 topics are derived from the outcome goals of the course/program and are rooted in the case-based pedagogy.**

Keywords—Software engineering, curriculum, case-based teaching, pedagogy

I. INTRODUCTION

A group of software engineering educators, committed to the advancement of the teaching of software engineering has embarked on a project to develop and support the creation and adaptation of a common set of learning materials and teaching strategies at the undergraduate level of education. The expected outcome of this project is a framework, in the form of a repository of essential software engineering teaching modules, assessment artifacts, course projects, and assignments that will be open to all educators in the discipline of software engineering. The philosophy of the research is that the availability of shared teaching material that have been developed by those knowledgeable in each topical area will advance the learning goals, and promote the adoption of teaching strategies that are in the students' best interest.

The goal of the research is the identification of commonalities in teaching software engineering across multiple territories and institutions and the identification of a fundamental set of topics, and course material for teaching software engineering at the undergraduate level. The researchers will bring expertise from multiple sub-disciplines to ensure that a broad spectrum of shareholder interests' are represented. The data gather from shareholders will result in an integrated environment that leverages a set of best practices, with respect to software engineering teaching techniques, processes, and material - towards the definition of an international collaborative teaching paradigm.

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.

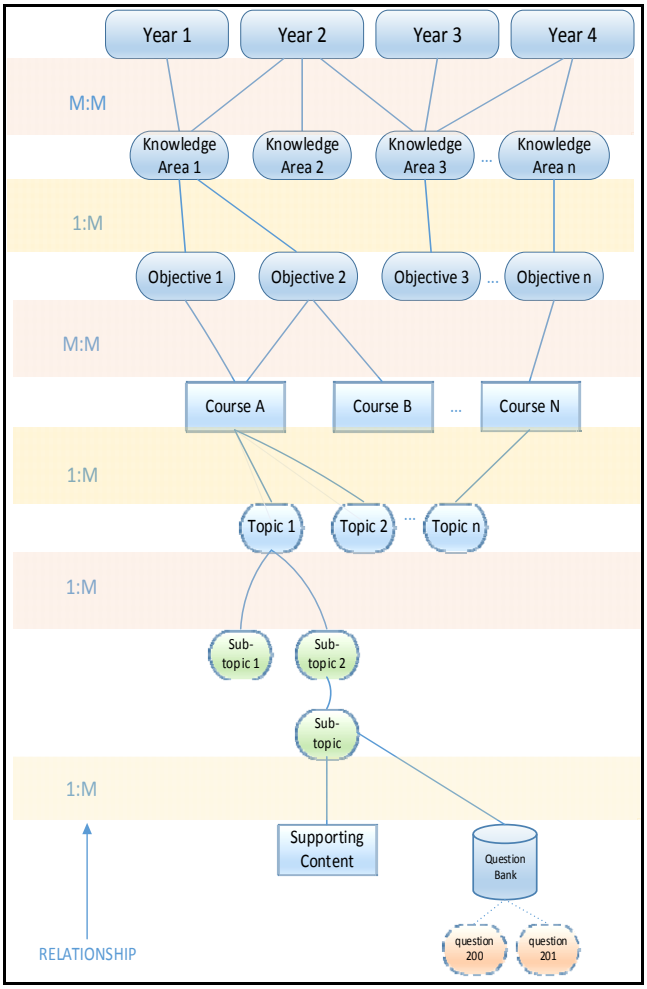

Figure 1. Repository Topography

A. *Case-based Pedagogy*

The workshop participants will examine the need for case-based teaching method in computing education and subsequently discuss the requirements for a case repository and set of practical tips on how case method can be incorporated into the different courses, over a four-year undergraduate degree in software engineering.

Faculty teaching computing courses face the difficult challenge of how to prepare students for the real world of computing practice. It is essential to teach concepts and principles but at the same time one has to translate this directly into real-world activity, for example, in software engineering discipline, how to design, develop, and maintain software solutions. Though some faculty have used case method for teaching computing courses, there has been no concerted effort by the professional computing communities to promote this pedagogy. Case method has been successfully applied in teaching other professions including law, medicine, and business [1]. Though there is great pedagogical value in case method, computing has not leveraged this method for teaching. One of the main reasons for this is the lack of cases and a repository where faculty can search and find useful cases. Another is the fact that there is very little written and sharing among faculty on how to use cases when teaching technology oriented topics such as software engineering, programming, solution architecture, etc.

B. *What is case method?*

The case method uses two elements namely the case, and a set of activities related to that case [1]. The case is a rich narrative that provides detailed information about a situation in which an individual or group must make a decision or solve a problem. Cases take many forms, and there are varieties of ways to write them. Usually it comprises the following:

- A detailed description of the problem's context, which at the least include the current situation and background information,
- A character that plays the central role and that character has to be solve the problem,
- Supporting data, this can include a range; data tables, quoted statements from the various actors in the case, supporting documents, images, video, or audio
- The case is not required to provide any analysis or conclusions.

Students working on the case during the classroom activities focus on analyzing the case to understand and explain the events, evaluate and propose solution options to solve the problem, predict the effects of taking actions, etc. The activities can be classified into the following four steps [2]:

1. Understanding the case-identifying the important facts of the case,
2. Analyzing the case by understanding the issues and challenges from multiple perspectives, evaluating solutions proposed in the case,
3. Taking action by proposing alternate solutions, and evaluating the pros and cons of the solutions and their short- and long-term impact, and
4. Finally, being able to "take away" the generalizable concepts and principles from the case.

The activities can differ depending on the course that is being taught. For example, in a management oriented course, an activity can be answering discussion questions pertaining to the case, "Should the marketing manager launch the product?". In a technology oriented course, an activity can include the students having to design and evaluate solutions, "Propose and evaluate two alternative IT solution architectures for implementing the automated fulfilment process".

C. *The Rationale for Case-based Teaching*

Case teaching method provides a number of benefits to enhance student learning through "interactive pedagogy" by stimulating critical thinking and problem solving skills and by creating reasonably realistic replicas of actual situations--which include incomplete information, time constraints, and conflicting goals. This leads to enhances student motivation and well-aligned learning environment, where practice and theory come together. This is especially true when teaching technical topics through traditional lecture that is often very dry and boring. Following are some benefits of using cases in computing education:

- Helps to introduce real world scenarios and problems into the classroom,
- Convey knowledge of what computing professionals do and how they work,
- Develop effective problem-solving skills, which are situated in a real world context,
- Helps students to better connect theory and practice,
- Enhance cooperative learning skills in the class,

D. *The Way Forward*

In order to drive the case method in computing education, as a community involved in teaching computing courses, we need to:

- Develop a set of sample cases that can be used in teaching specific computing courses, for example, object oriented design, enterprise integration, software testing, etc., and share them with other faculty.
- In order to enable sharing, we need to develop an online repository where faculty can submit teaching cases and also search and download teaching cases.

To drive the above actions, we intend to run Case Method for Computing Education (CMCE) workshops in major computing education conferences. An important aspect of any research is the use of data that will be transformed into information through the process of the research work. It is fundamental that the data used is correct and sufficient. It is expected that the workshops used for data capture in this

research will correct and sufficient, as it will be sourced from a diverse population in multiple sessions.

II. THE RESEARCH PROJECT

Figure 2 illustrates the activities of the first phase of the this project. The pre-Repository seminar was the first event, and had the stated goals of disseminating information, gathering initial opinions, and inviting participation. The second event is a series of workshops that are intended to garner research data, and establish collaborative relationships with a broad range of researchers, private sector representatives, and government agencies at various institutions.

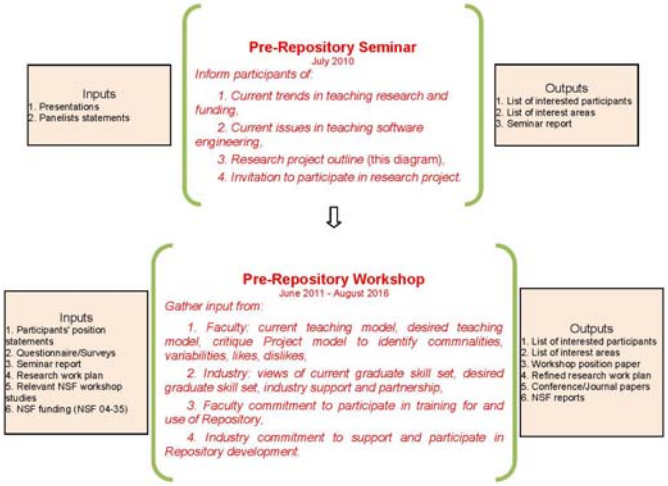

Figure 2. Research Project Phase 1 Description

The 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 [3] examined the funding and research trend in SE in the USA, and looked at the impact the collaboration between Asian researchers and their United States counterparts may have on the research trends. The second presentation [4] looked at a novel approach to teaching object-oriented analysis and design as a topic of a SE course. The hands-on tutorial [5] 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.

Figure 3 outlines 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, case studies, etc.), and course assessment material (student surveys/questionnaires, course evaluations, etc.).

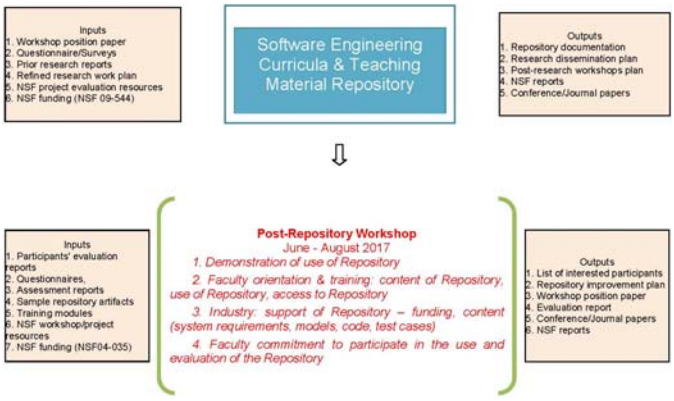

Figure 3. Research Project Phase 2 Description

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 [6]. Cloud Computing, software as a service (SaaS) [6], and service-oriented architecture (SOA) [7] 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.

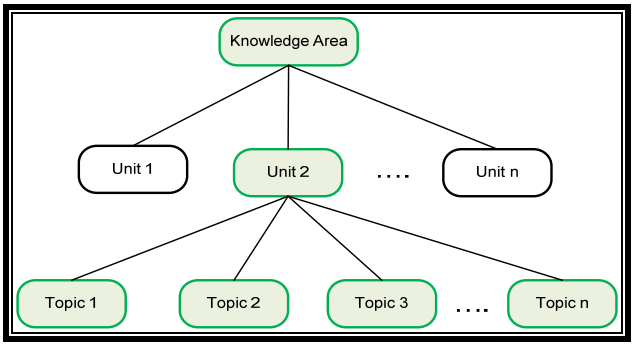

Figure 3. SEEK hierarchical organization showing KAs, Units and Topics

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[8]. 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 will build on the work already done by Grant, *et. al*. in [9] and [10] 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.

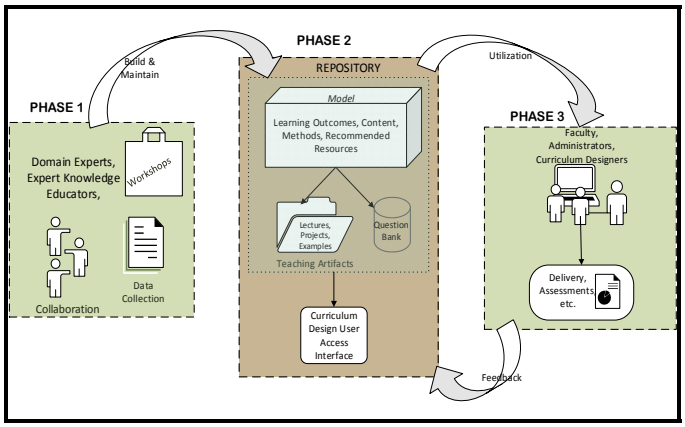

Figure 5. Overview of three phase structured approach

III. PHASE 1 WORKSHOP DESCRIPTION

The workshops call for submissions will require participants to submit position papers that outlines their specific interest, anticipated contribution, and expected outcome of the workshop. Participants will be informed of the expected outcome of each breakout panel session and the workshop in general. Plenary sessions will have presentations made by participants that represent the focus of each breakout session group. At the end of the presentations there will be a question and answer period. The presenting authors of these selected submitted papers will be asked to co-chair the relevant breakout session panel.

The main input used for the breakout panel sessions will be sections from the Institute of Electrical and Electronic Engineering Computer Society (IEEE-CS) and Association of Computing Machinery (ACM) Computing Curriculum, Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science (IEEE-CS/ACM CS 2013) [8] document. The selected section of the document will be distributed to each of the participants, prior to the start of the workshop for their review and preparation. The selected sections of the document will comprise introductory information on the IEEE-CS/ACM CS 2013 manuscript, and description of the SEEK, along with tables of the SEEK areas, with the supplementary information.

It is planned that each of the panel groups will be comprised of four to six members. This would make for a minimum of twenty to a maximum of thirty workshop participants, plus three workshop organizers and one to two resource personnel. The moderator's main task will be the recording of all panel matters compiling the panel report. Each panel will have a moderator, who will be a project researcher or one of the presenting authors from the plenary session. Each panel will also include students, possibly both international and local.

The main activities of the workshop will be the breakout panel sessions and the wrap-up plenary session. For the breakout, panel sessions the participants will be divided into five 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,
- Software Process and Software Quality, and
- the additional topic of case-based teaching.

Each panel group will discuss the topics listed under the respective heading, from the SEEK areas. The groups will answer the following questions:

- Is this topic relevant to the fundamentals of SE?
- In which year(s) of the program should it be taught?
- What percentage of teaching time should be allocated?
- What is the rationale for the selection of this topic?

In the wrap-up session of the workshop, the co-moderators will present the findings of their respective panels. These presentations will be followed by a question and answer session. This session is intended to gather final overall comments and recommendations and get the participants feedback on their views on the success of the day's activities. Specifically, the research group will seek to determine what the participants thought was most useful, undesired, what should have been done more/less, and what influence the sessions would have on their current teaching strategies.

IV. RELATED WORKS

The Software Engineering Disciplinary Commons (SEDC) (http://sec.cs.siue.edu/) is a project that is funded by the National Science Foundation (NSF) (http://www.nsf.gov) of the United States, and the program has two 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 computer science in general) 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.

This work encompasses that of the SEDC, and adds three more layers; namely – (1) international participation, (2) collaborative distributed teaching, and (3) retrieval and update of teaching material. Two of the researchers involved in this project are participants in the SEDC project, and their experience from the SEDC project will be incorporated into this project.

The IEEE Computer Society hosted a series of workshops in the early 1980s that initiated work on establishing standards for software engineering (SE) products and processes. That effort motivated the establishment of a Software Engineering Body of Knowledge (SWEBOK), which came into focus ten years later. The mission statement of SWEBOK is "To establish the appropriate sets(s) of criteria and norms for professional practice of software engineering upon which industrial decisions, professional certification, and educational curricula can be based." This work is currently embodied in the 2004 SWEBOK Guide that is comprised of twelve chapters and four appendices. The guide sets out a baseline for body of knowledge of SE. The IEEE Computer Society seeks to keep the Guide relevant by updating it based on comments it receives from educators and practitioners in SE.

Sections of this document will be used in the workshop, as foundational starting points for the work to be accomplished in the workshops. The goal is to build from the existing body of knowledge in this research area, towards achieving the project goals. Both IEEE documents are currently under-going major updating and an outcome of the workshop effort is in contributing to this update activity.

REFERENCES

[1] Williams S. M., Putting Case-based Instructions into Context: Examples from Legal and Medical Education., Journal of Learning Sciences, vol2, 4, 1992, pp.367–427.

[2] Shulman, J. H., Happy Accidents: Cases as Opportunities for Teacher Learning. American Educational Research Association April 2002, New Orleans, USA.

[3] Grant, E. S. (2010). Trends in Software Engineering Research, International Seminar on Software Engineering Development and its Application in Education, Holy Angel University, Philippines.

[4] 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..

[5] Salta, M. T. (2010) Online Courseware Development, International Seminar on Software Engineering Development and its Application in Education, Holy Angel University, Philippines.

[6] 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

[7] 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.

[8] Joint Task Force on Computing Curriculua, ACM/IEEE-CS, Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, ACM, IEEE-CS USA, Dec. 2013.

[9] 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.

[10] 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.