Using the case method to transform engineering education

Massood Towhidnejad, Thomas B. Hilburn, Salamah Salamah

ABSTRACT
One of the primary objectives of most undergraduate engineering programs is to prepare graduates for engineering practice. Advances in engineering methods and technology, modern system complexity, and the need to work across cultures and time zones have influenced significant change in the practice of engineering. Unfortunately, there has been little change in engineering education to accommodate these changes. In this paper, we propose that engineering educators alter the typical lecture style and instructional activities focused on the individual, and use a team-based case method approach to transform their programs, so that throughout a curriculum teams of students engage in realistic engineering practice activities, centered around a comprehensive life-cycle engineering case study. The authors illustrate this approach with a life-cycle software engineering case study.

Keywords: case study teaching, engineering education, software engineering
1. INTRODUCTION

Advances in engineering methods and technology, modern system complexity, and the need to work across cultures and time zones have influenced significant change in the practice of engineering. The same cannot be said of engineering education: for over 100 years curricula have been packaged in a course-by-course, semester-by-semester framework; material is mostly delivered by lecture and individual homework assignments; classrooms are characteristically arranged in rows and columns of single-person desks; course grades are generally assigned on the basis of individual work; faculty team teaching is rare; significant team projects are typically reserved for senior level capstone courses; many faculty lack experience in professional practice and team building, and have little appreciation for the role of general education in engineering education.

A recent study, involving an engineering competencies survey of over 4,000 alumni, in eleven different engineering disciplines, showed that the four highest rated competencies were the following: “ability to function on a team”, “engineering problem-solving skills”, “ability to analyze and interpret data”, and “written and oral communication skills”. In this paper, the authors discuss how a case study approach, using a “life-cycle case study”, can be effective in teaching engineering competencies and potentially transforming the way an engineering curriculum is delivered.

2. CASE STUDY TEACHING

The case method is a teaching approach that involves presenting students with a case study, and putting them in the role of a decision maker facing a problem. The case study typically describes a problem, using a scenario format to provide context and to summarize key issues and events related to the problem. The scenario might be supplemented with background material (setting, personalities, sequence of events, and problems and conflicts), artifacts, and data, which is relevant to the situation depicted. Students then engage in study, analysis, and discussion to solve the problem.

The literature describes a variety of ways that the case method can be used to support learning and research. Case studies should be viewed as active learning tools; they are meant to encourage participation, debate and understanding. Although they can be used in a didactic, teacher-centered pedagogy, they are most effectively used in an active learning, student-centered system where the teacher acts as a facilitator. The case method can be mixed and matched with other pedagogies such as lectures, guided discussions, and project work. Case studies can be used to supply the background needed for specific problems and design projects; serve as subjects for class discussions; or they can be used to motivate further study, and to identify and formulate research problems.

3. THE CASE METHOD FOR ENGINEERING

Although the case method has proven its worth and is a widely used method of teaching in fields such as business, law, and medicine, it has not been accepted to the same extent in engineering education, except in a senior project design course. ABET requires that all engineering programs involve their students in “a major design experience based on the knowledge and skills acquired in earlier course work and incorporating appropriate engineering standards and multiple realistic constraints”. Students are typically grouped into teams to work on a semester or year-long “realistic” engineering development project. Unfortunately, this is too often somewhat isolated from the rest of the curriculum and does not form a real-world basis for the entire curriculum. Although, students may be exposed to elements of engineering practice in their foundation engineering courses, they often arrive in their senior project course with disjointed course view of engineering practice and have an insufficient appreciation and understanding of the complexity and multifaceted nature of real-world development environments. In addition, students often lack sufficient team, analysis, and communication skills. Therefore, it is imperative that engineering curricula introduce professional and real-world education throughout a curriculum. Many programs now introduce small team-based engineering projects in the early part of the curriculum. Case studies can be used to support and extend such experiences.

An excellent way to address the challenges of introducing real-world exposure through a curriculum is the use of a robust life-cycle engineering case study – that is, a case that engages the student in the engineering of a complex system throughout its “life”: system definition, preliminary and detailed design, implementation, verification and validation, and operation and maintenance. Of course the system chosen would vary with the engineering field: for example, an airplane for aeronautical engineering, a suspension bridge for civil engineering, a power grid system for electrical engineering, or
a smart house system for software engineering. Friedman and Sage present a technique for analyzing and assessing the life-cycle features of a system engineering case. We discuss an example of the life-cycle engineering case study approach in the next section.

4. THE DIGITALHOME CASE STUDY

For some time the authors have been involved in a "smart house" case study project that focuses on a DigitalHome (DH) system. The DH project, when completed, will cover the life-cycle development of a software product (project management, requirements analysis and specification, design, implementation, testing, and maintenance).

The DH project is based on a scenario about a real-world, but fictitious, company, HomeOwner, which is the largest national retail chain serving the needs of home owners. Homeowner, based on market and technology research, decides to invest in the development of a "smart" house, the DH system, which will have the following features:

- The DH system shall allow any web-ready computer, cell phone or similar device to control a home's temperature, humidity, lights, security devices, and the state of small appliances.
- The communication center of the DH system shall be a personal home owner web page, through which a user can monitor and control home devices and systems.
- The DH shall contain a master control device that connects to the home's broadband Internet connection, and uses wireless communication to send and receive communication between the DH system and the home devices and systems.
- The DH shall be equipped with various environment sensors (temperature sensor, humidity sensor, power sensor, contact sensor, etc.). Using wireless communication, sensor values can be read and saved in the home database.
- The DH system shall include programmable devices (thermostats, humidistats, contact sensors, and small appliance and lighting power switches), which allows a user to easily monitor and control a home's environmental characteristics from any location, using a web ready device.
- The DH system shall include a DH Planner, which provides a user with the capability to direct the system to set various home parameters (temperature, humidity, and on/off appliance and lighting status) for certain scheduled time periods.

The initial phase of the case study project has concentrated on building a foundation for full development: research into case study teaching; identifying a case study problem; creating a scenario framework; describing the launch of the software development team; fashioning a software development plan to guide development of the DH system; establishing a development process; creation of a DH need statement; analysis, modeling and specification of the DH requirements; development of a system test plan; development of a software architecture; and specification of the system components.

A set of DH mini-case studies (that we call "case modules") have been created; they are designed to engage students in active learning software engineering activities related to the DH system. With this life-cycle engineering case study, we advocate a "Team Learning Format". Our approach involves the following activities:

- Students are assigned prior reading or other preparation.
- The instructor introduces the case, providing motivation and giving background.
- The instructor divides the class into teams, and if fitting, assigns roles.
- Teams work on a case module exercise: discussing problems/issues, answering questions and making decisions, and prepare a report of their conclusions.
- Teams present their report to the class, and the class discusses results.

Existing DH scenarios, artifacts, and case modules can be viewed at http://www.softwarecasestudy.org/. The case study material will eventually include a complete set of software development artifacts as well as case exercises that can be used to teach different topics (i.e., requirements, design, programming, testing, quality reviews, project management, etc.) throughout a computer science or software engineering curriculum. Each case exercise represents a mini case study and is associated with a specific teaching subject (e.g. project planning, requirements inspection, object oriented design, system test planning, etc.) and a set of learning objectives.
5. ENGINEERING CASE MODULES

An engineering case module is a mini case study, extracted from a comprehensive life-cycle engineering case study, which can be used to engage students in activities that teach them about professional engineering practice. The DH case study includes three general types of case modules:

- **Type 1** – Case modules that involve solving an engineering problem by creating an engineering artifact: for example, a project budget and schedule, an analysis model, a design architecture, or a plan for testing an engineered product.
- **Type 2** – Case modules focusing on review of engineered artifacts: for example, a formal preliminary design review or a peer review of the detailed design for a system component.
- **Type 3** – Case modules that motivate discussion, assessment, and decision related to non-technical problems: for example, a team personnel issue or an ethical or professional issue for an individual engineer.

As an example of a Type 1 DH case module, the module entitled “Developing a Project Plan” asks students to create a plan, which includes estimates of the required project effort, the project duration, and identifies and schedules project tasks, and assigns the lead for each task. The DH case module “SRS Inspection” is a Type 2 case module; it is covered in detail in the next section. The DH module, “Team Problems”, illustrates a Type 3 module. The module presents a team of students with a scenario about team personnel problem and asks them to decide on various approaches to solve the problem. The case module scenarios concern problems such as conflicts between individual members (“Double Trouble”), leadership effectiveness (“There’s gonna be some changes made”), professional behavior (“Don’t worry your pretty little head”), and appropriate work ethic (“Slacker on the loose”).

The authors believe many of the DH life-cycle case modules could be adapted to other engineering disciplines. Activities involving analysis and design, project planning, peer reviews and product verification, and solving team problems are relevant to all fields of engineering. Such activities are particularly effective when they are tied together by a single self-study, used over a variety of topics and courses. This approach allows students to engage in whole range of activities related to a complex engineering project, which has the effect of providing realism and continuity to a curriculum.

6. A DIGITALHOME CASE MODULE

An early part of the Digital Home case study effort was the analysis and specification of the software requirements for the system, which resulted in a DH Software Requirements Specification (SRS). In addition to the specification of the functional requirements, the SRS includes a description of user characteristics, development constraints, the performance environment, and nonfunctional requirements specifying performance, reliability, and safety and security requirements. The SRS also includes a use case model. As part of the development of the requirements specification, the authors developed a case module titled “SRS Inspection”. A software inspection is a systematic peer examination of a software artifact with the purpose of detecting and identifying software product defects. The counterpart to a software inspection in other engineering professions is the engineering “design review”.

The learning objectives for the case module are to provide student increased ability to:

- Work as a member of an inspection team.
- Assess the quality of a SRS.
- Describe problems in specifying the requirements for a software product.
- Work more effectively as part of a team.
- Explain the inspection process.
- Describe how inspection data can be used to assess the quality of a software artifact and the effectiveness of an inspection.

The learning objectives for this case module go beyond assessing the quality of the SRS, but are intended to address critical software engineering education goals: appreciating and understanding the problems in specifying requirements; learning to work as part of a team; using and following an inspection process; and analyzing inspection data. These objectives are consistent with the most valued ABET outcomes mentioned earlier. Similar types of learning objectives are part of most of the DH case modules.
Students are provided a set of DH artifacts (a DH Background Scenario, DH Team Biographical Sketches, a Customer Need Statement, a High Level Requirements Definition, and the DH SRS, Version 1.2) and an Inspection Package (an Inspection Process Description, a SRS Inspection Guidelines and Checklist, a Defect Log, an Inspection Summary Report Spreadsheet, and an Inspection Summary Report Spreadsheet Example). Although the DH SRS, Version 1.2 artifact is the chief focus of the inspection, the other material (such as the background scenario and need statement) provide the setting and context to create a realistic environment for a professional and effective software inspection exercise. The inspection package provides the sort of forms and tools used in a mature inspection process and help guide the students not only in performing an effective inspection, but in understanding how this “best practice” works.

Each case module contains a scenario to provide background, context, and some realism to the learning activities associated with the case module exercise. Table 1 contains the scenario that is part of the SRS Inspection Case Module. Each case module also provides a set of teaching notes intended to support use of the case module in undergraduate and graduate software development courses, providing guidance and suggestions to a teacher using the case module. Table 2 contains some of the teaching notes for SRS Inspection module.

Table 1. SRS Inspection Module scenario.

<table>
<thead>
<tr>
<th>Case Study Participants:</th>
</tr>
</thead>
<tbody>
<tr>
<td>† The DH Team &amp; Jose Ortiz, Director, DigitalHomeOwner Division of HomeOwner, Inc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario:</th>
</tr>
</thead>
<tbody>
<tr>
<td>In early September of 2010, HomeOwner Inc. (the largest national retail chain serving the needs of home owners) established a new DigitalHomeOwner division that was set up to explore the opportunities for equipping and serving “smart houses” (dwellings that integrate smart technology into every aspect of home living). The Marketing Division produced two documents: the DH Customer Need Statement and the DH High Level Requirements Definition (HLRD). In September 2010, a five person team was assembled for the project and in early October 2010 carried out a “project launch”. After project planning was completed the team began work on requirements analysis and specification. The first version, 1.0, was completed in early October and versions 1.1 and 1.2 were completed by mid-October. In consultation with Jose Ortiz, the team has decided to carry out a formal Fagan inspection of the SRS, version 1.2. Jose has agreed to act as a customer on the inspection team, Michel Jackson is the author, Disha Chandra will be the moderator and other roles will be assigned in the overview meeting.</td>
</tr>
</tbody>
</table>

Table 2. SRS inspection Module teaching notes.

1. This case module could be used in different level courses (from an introductory level course in software engineering to an upper level or graduate course in requirements engineering or quality assurance.).
2. Assuming an adequate student preparation for the exercise, allowing students about three hours each for the exercise should be sufficient: assuming two hours for preparation and inspection, and one hour for the inspection meeting.
3. Preparation time can be done as an outside class activity to be reported as part of a deliverable before inspection meeting.
4. It would be beneficial to follow the exercise with a twenty to thirty minute discussion concerning the student team results. Some key points to include in the discussion are the following:
   5. Discuss how closely the inspection process was followed:
      † How well did the team conduct each phase?
      † How well did students carry out their assigned inspection role (i.e., moderator, author, inspector)?
      † …

6. Student team members should be cautioned about a few things:
   † Leave their ego outside of the meeting room
   † Their job is to identify defects, not fix them.
   † …
to identify missing features. In addition, the team, problem-solving and analysis skills were enhanced. We should also note that the collection and analysis of team inspection data provides the teacher with excellent information on how the inspection was conducted and the degree to which the case module objectives were reached.

7. A CURRICULUM-WIDE CASE METHOD

Table 3 presents a framework for coverage of various software engineering practices throughout a software engineering curriculum. The courses are those described in a reference curriculum developed by the Association Computing Machinery and IEEE Computer Society.12

Each of case study areas, listed in Table 3, would have at least one DH software artifact and one or more case modules. For example, in the area of Software Requirements Specification, the project currently has the following material developed or in progress:

- Customer Need Statement
- Needs Assessment Case Module
- Several versions of the DH SRS (pre and post inspection)
- SRS Inspection Case Module, with a package of inspection support forms and tools.
- Requirements Change Case Module
- Use Case Model
- Use Case Inspection Case Module
- Use Case Modification Case Module
- Operational Profile Case Module

Table 3. A DigitalHome curriculum framework.

<table>
<thead>
<tr>
<th>Case Study Area</th>
<th>Course(s)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Planning</td>
<td>SE323 Software Project Management</td>
</tr>
<tr>
<td>Software Process Description</td>
<td>SE324 Software Process and Management</td>
</tr>
<tr>
<td>Software Requirements Specification</td>
<td>SE322 Software Requirements Analysis</td>
</tr>
<tr>
<td>User Interface Specification</td>
<td>SE212 Software Engineering Approach to HCI</td>
</tr>
<tr>
<td>Traceability Matrix</td>
<td>SE322 Software Requirements Analysis</td>
</tr>
<tr>
<td>Architectural Specification</td>
<td>SE311 Software Design and Architecture</td>
</tr>
<tr>
<td>Module Specification</td>
<td>SE311 Software Design and Architecture</td>
</tr>
<tr>
<td>Algorithm Design Specification</td>
<td>SE211 Software Construction</td>
</tr>
<tr>
<td>Coding</td>
<td>SE211 Software Construction</td>
</tr>
<tr>
<td>Unit Test Planning</td>
<td>SE211 Software Construction</td>
</tr>
<tr>
<td>Integration Planning</td>
<td>SE311 Software Design and Architecture</td>
</tr>
<tr>
<td>System Test Planning</td>
<td>SE321 Software Quality Assurance and Testing</td>
</tr>
<tr>
<td>All of the Above</td>
<td>SE101 Introduction to Software Engineering, SE400 Software Engineering Capstone Project</td>
</tr>
</tbody>
</table>

* Course number and names are from Diaz-Herrera and Hilburn.12

The authors have used the case modules in a variety of computing courses and workshops: undergraduate and graduate courses in software engineering, a workshop in software reliability for faculty and researchers, and a faculty workshop in case study teaching. These experiences support the belief that engineering practices are best taught by applying the case method approach across all phases of a lifecycle case study.

8. CONCLUSIONS

Although there is still work to be done to complete the DH case study, we hope the material developed and our experiences with it provide insight into the value of such an approach. We believe the content,
organization, and spirit of the DH Case Study provides a model for the development of other engineering life-cycle case studies. We are encouraged by our experience that such an approach can transform the education of engineering students, teaching them engineering practice through real or quasi-real case study exercises. As a minimum, the DH can easily serve as a baseline for case studies/modules for Computer Engineering, Software Engineering, and System Engineering. Hopefully, this discussion will broaden the reach of this work into other fields of engineering with a goal of building a community of collaborators that will contribute more significantly to the development of case studies, artifacts, and exercises.

Acknowledgements
Initial work on the DigitalHome case study was funded as part of the NSF project: “The Network Community for Software Engineering Education” (SWENET) (NSF 0080502). In addition, the current work on the case study is funded through NSF’s (DUE- 0941768) “Curriculum-wide Software Development Case Study”.

REFERENCES