
AC 2011-1690: REPORTING ON THE USE OF A SOFTWARE DEVELOPMENT CASE STUDY IN COMPUTING CURRICULA

Massood Towhidnejad, Embry-Riddle Aeronautical Univ., Daytona Beach

Massood Towhidnejad is a tenure full professor of software engineering in the department of Electrical, Computer, Software and System Engineering at Embry-Riddle Aeronautical University. His teaching interests include artificial intelligence, autonomous systems, and software engineering with emphasis on software quality assurance and testing. He has been involved in research activities in the areas of software engineering, software quality assurance and testing, autonomous systems, and human factors.

Thomas B Hilburn, Embry-Riddle Aeronautical Univ., Daytona Beach

Dr. Thomas B. Hilburn is a Professor Emeritus of Software Engineering at Embry-Riddle Aeronautical University. He has also worked on software engineering research and education projects with the FAA, General Electric, Harris Corp, the MITRE Corporation, DOD, FIPSE, the SEI and the NSF. His current interests include software processes, object-oriented design, formal specification techniques, and curriculum development. He is an IEEE Certified Software Developer, SEI-Certified PSP Developer, and currently chairs the Curriculum Committee of the IEEE-CS Educational Activities Board and Planning Committee of the IEEE-CS Professional Activities Board.

Salamah Salamah, Embry-Riddle Aeronautical Univ., Daytona Beach

Reporting on the Use of a Software Development Case Study in Computing Curricula

Abstract:

The use of case studies is an effective method for introducing real-world professional practices into the classroom. Case studies have become a proven and pervasive method of teaching about professional practice in such fields as business, law, and medicine. The term “case study” is used in a variety of ways. In its most naive form, it simply refers to a realistic example used to illustrate a concept or technique. Although the use of case studies in education has shown success in the above mentioned disciplines, it is yet to be adopted in any significant way in the computing education. In this work, we re-introduce the DigitalHome Case Study, and report on our use of one of the case modules of this case study.

1. Introduction to Case Study Teaching

Case studies were first used in the Harvard Law School in 1871 [2]. Since then, case studies have been a subject of much study and research about their effectiveness in teaching and learning [1,2,4,5]. They have become a proven and pervasive method of teaching about professional practice in such fields as business, law, and medicine. In its most naive form, case study teaching refers to a realistic example used to illustrate concepts and/or technique. More formerly, a case study involves the application of knowledge and skills, by an individual or group, to the identification and solution of a problem associated with a real-life situation. Such a case study would contain an account (often in a scenario format – a story) of a real-world activity, event, or situation. It might be supported with background material (setting, personalities, sequence of events, and problems and conflicts), artifacts, and data, which is relevant to the situation depicted.

Although case studies can be used in a didactic, teacher-centered pedagogy they are most effective when used in an active learning and student-centered system where the teacher acts as a facilitator or coach. Case studies are of special value in problem-based learning, which concentrates on the development of problem-solving skills, self-directed learning skills, and team skills. There are a number of excellent examples of case studies in science and engineering at the SUNY-Buffalo web site [12]

Although the use of case studies in education has shown success in the aforementioned disciplines, it is yet to be adopted in any significant way in the computing education. One of the reasons for the lack of use of the case-study approach is the shortage of sufficient material for this purpose. There are many software textbooks that use case studies (as examples to illustrate concepts and techniques): [10] (several, including an Airline Reservation System and a Household Alarm System), [11] (Fireworks Factory), [13] (Picadilly Television and Ariane-5), [15] (SafeHome). Additionally, there has been an increased effort in introducing case studies in computing courses as reported in the literature [9, 14].. These case studies often lack the following:

- Realistic artifacts (often space does not allow providing a complete requirements or design document)

- Completeness (covers only a portion of the life-cycle, and not an end-to-end), with a focus on design and implementation
- Ability to decouple from the text and apply in ways not intended by the author
- Techniques for integration into course activities or into the curriculum as a whole
- A scenario format that would motivate students to get engaged in problem identification and solution.
- Guidance to the instructor on how to use the case study to teach a course topic or concept

In previous work, the authors introduced the idea of a comprehensive case study (the *DigitalHome* case study) [7, 8] that can be used throughout a computing curriculum. In that work we motivated the use of case studies in teaching and shared some of the artifacts developed at the time. We also discussed how the *DigitalHome* case study addresses the aforementioned shortcomings with a goal of providing a complete set of artifacts associated with software development as well as providing case modules that can be used by faculty in teaching different subjects in a computing curriculum.

The main aim of the current paper is twofold; we provide an update on the new *DigitalHome* case study material developed in the last two years as part of a National Science Foundation grant. In addition, we highlight our experiences in using the case study material in different software engineering courses at our institution. We also discuss the future goals, both short and long term, of the case study project and the anticipated improvements and new artifacts.

2. The DigitalHome Case Study

The Case Study Project focuses on developing a complete set of artifacts associated with software development (e.g., Requirements Document, System Test Plan, ...) as well as case modules (a “mini-case studies”). These artifacts and case modules are related by being part of and derived from a single case, the development of a single software product. Each case module relates to an artifact or activity involved in the development of the product. In addition, each case module is framed as part of a product development narrative, using a scenario format, which involves characters and incidents that would be part of an actual software development project (e.g., formation of a software project team, interaction with upper management, customer and user interviews, writing a use case description, formal inspection of a software artifact, designing a class interface, a design walk-through, system testing, etc.).

The *DigitalHome* Case Study is intended to cover the complete life-cycle development of a software product (project management, requirement analysis and specification, design, implementation, testing and maintenance). The initial phase of the case study project, concentrated on building a foundation for full development: research into case study teaching; identification of a case study problem; creation of a scenario framework; description of a launch of the software development team; development of a software development plan to be used as part of the case study; and development of several related case modules.

3. The DigitalHome System

The *DigitalHome* project is part of the vision of the future for a national company HomeOwner, which is the largest national retail chain serving the needs of home owners in building,

furnishing, repairing, and improving their homes. The HomeOwner management has decided to develop a prototype DigitalHome (DH) system that has the following features:

- The DH system will allow any web-ready computer, cell phone or PDA to control a home's temperature, humidity, lights, and the state of small appliances.
- The communication center of the DH system will be a personal home owner web page, through which a user can monitor and control home devices and systems.
- The Digital Home will 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 Digital Home will be equipped with various environment sensors (temperature sensor, light sensor, humidity sensor, power sensor, contact sensor, water sensor, etc.). Using wireless communication, sensor values can be read and saved in the home database.
- The DH system includes programmable devices (thermostats, humidstats, and small appliance and lighting controllers), 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 includes a DH Planner, which provides a user with the capability to direct the system to set various home parameters (temperature, humidity, lighting, and on/off appliance status) for certain scheduled time periods.

4. Digital Home Material

The development of the DH cases study has consisted of writing scenarios, developing DH artifacts, case modules and exercises. Thus far, Project Inception, Project Launch and Planning, and Requirements Analysis and Specification have been completed. Table 1 provides a listing of the current artifacts, case modules, and exercises produced as part of the Case Study Project. The case study material is available for download and use at: <http://www.softwarecasestudy.org/>.

Table1. Current Case Study Material

Item	Development Phase
Beginning Scenario	Pre Project
DigitalHome Development Team Bios	Pre Project
Development Strategy	Pre Project
Customer Need Statement	Pre Project
Case Module: Assessing Needs	Pre Project
Exercise: Assessing Needs	Pre Project
Launch Script	Launch
Launch Scenario	Launch
Conceptual Design	Launch
Context Diagram	Launch
Development Process	Launch
Case Module: Software Process	Launch
Exercise: Software Process	Launch
Artifact: SRS 1.2	Requirement Analysis
Artifact: SRS 1.3	Requirement Analysis

Use Case Model	Requirement Analysis
Case Module: SRS Inspection	Requirement Analysis
Inspection Process	Requirement Analysis
Inspection Defect Log	Requirement Analysis
SRS Checklist	Requirement Analysis
Inspection Summary Report	Requirement Analysis
Exercise: Requirements Inspection	Requirement Analysis
Case Module Operation Profile	Operation and Maintenance
Exercise: Creating an Operation Profile	Operation and Maintenance
Case Module: Software Team Problems	Others
Exercise: Software Team Problems	Others

4.1 Software Requirements Inspection Material

The following section describes the Case Study module and exercise material associated with requirement inspection. Table 2, shows the software inspection case module, while Table 3 shows the associated exercise.

Table2. Requirement Inspection Case Module

Case Module : SRS Inspection
Prerequisite Knowledge: Understanding of basic elements of a Fagan Software Inspection process.
<p>Learning Objectives: Upon completion of this module students will have increased ability to:</p> <ol style="list-style-type: none"> 1. Work as a member of an Inspection Team 2. Assess the quality of a Software Requirements Specification(SRS) 3. Describe problems in specifying the requirements for a software product. 4. Work more effectively as part of a team. 5. Explain the inspection process. 6. Describe the value of the Fagan inspection process.
<p>Case Study Artifacts: DH Customer Need Statement, DH High Level Requirements Definition (HLRD), DH Background Scenario, DH Team Biographical Sketches, DH SRS, Version 1.2</p> <p>Inspection Package:</p> <ul style="list-style-type: none"> • Inspection Process Description • SRS Checklist • Defect Log • Inspection Summary Report Form
<p>Appendices: Exercise Booklet</p>
<p>Resource Information:</p> <ul style="list-style-type: none"> • [Fagan,1976] Fagan, M.E., Design and Code inspections to reduce errors in program development, 1976, <i>IBM Systems Journal</i>, Vol. 15, No 3, Page 182-211. (http://www.mfagan.com/ibmfagan.pdf) • [Fagan, 1986] Fagan, M.E., Advances in Software Inspections, July 1986, <i>IEEE Transactions on Software Engineering</i>, Vol. SE-12, No. 7, Page 744-751.

<http://www.mfagan.com/aisi1986.pdf>

- [Humphrey 2000] Humphrey, Watts S., *Introduction to the Team Software Process*, Addison-Wesley, 2000.

Teaching Notes:

- This case module could be used in different level courses (from a software level introductory course in software engineering to an upper level or graduate course in requirements engineering or quality assurance.).
- 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.
- 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:
 - 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)?
 - What was the quality of the inspection outline (NOTE: it is very helpful if the instructor complete the product inspection, and compare the students outcome to the instructor outcome)
 - How well was time managed?
 - Were all forms completely correctly and all appropriate data collected (number of defects, time spent for inspection, etc.)?
 - Address any personal conflict or egoistic attitude displayed.
- Student team members should be cautioned about a few things:
 - Their job is to identify defects, not fix them.
 - The SRS is a Requirement document and not a Design specification; so inspect appropriately and check that design elements have not been introduced by the author or as a result of inspection. This does not include design constraints specified by the customer.
 - Come prepared to the inspection meeting – students need to spend appropriate effort during their individual inspections.
 - It is critical that students understand the data they need to collect (time and defect data), how to use the forms provided, and the importance being careful and accurate in collecting and recording data. If students do not properly collect and record inspection data it will be difficult to assess the effectiveness of the inspection.
- If the course involves actual student development teams, this exercise could provide a good team building experience: it could be carried out near beginning of a course; it does not require deep technical knowledge; and everyone can participate.
- If the time and the size of the class permits the course instructor could take on the role of Jose Ortiz.
- If the class is divided to multiple teams, if possible, it is best to have each team conduct their inspection in a separate room, thus eliminating distracting noise generated by other teams.
- It would be best to hold the inspection meeting in class, since it would allow the instructor to observe this part of the inspection process.

Table3. Requirement Inspection Exercise

<p>Scenario</p> <p>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). In September and October of 2010, the Marketing Division of HomeOwner conducted a needs assessment for a DigitalHome product that would provide the computer and communication infrastructure for managing and controlling the “smart” devices into a home to best meet the needs and desires of homeowners. The Marketing Division produced two documents: the DH Customer Need Statement and the DH High Level Requirements Definition (HLRD).</p> <p>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 February and versions 1.1 and 1.2 were completed by mid February.</p> <p>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, Sumeera Nangia will be the moderator and other roles will be assigned in the overview meeting.</p>
<p>Learning Objectives:</p> <p>Upon completion of this case module students will have increased ability to:</p> <ul style="list-style-type: none">• Work as a member of an Inspection Team• Assess the quality of a Software Requirements Specification(SRS)• Describe problems in specifying the requirements for a software product.• Work more effectively as part of a team.• Explain the inspection process.• Describe the value of the Fagan inspection process.
<p>Reading Assignment</p> <p>Read the below Case Study Artifacts:</p> <ul style="list-style-type: none">• DH Background Scenario• DH Team Biographical Sketches• DH Customer Need Statement• DH High Level Requirements Definition (HLRD)• DH SRS, Version 1.2
<p>Exercise Description</p> <ol style="list-style-type: none">1. As preparation for the case module, read the Case Study Artifacts listed above, except for the DH SRS. The DH SRS will be reviewed as part of the inspection process in part 3 below.2. You will be assigned to a small development team (like the DH team).3. Your team is to take on the role of the DH Team and prepare for the inspection of a Software Requirements Specification.<ol style="list-style-type: none">a. First, review the inspection package materials (list above).b. Then follow the Inspection Process and any special instructions from your teacher.4. <u>Note:</u> The Rework and Follow-Up phases of the Inspection Process are not part of this exercise. Also, the part of the Exit Criteria “Software work product has been revised, removing identified defects” does not apply in this exercise.

5. Use of DigitalHome Inspection Case Module and Exercise

Over the past seven years, the authors has used case studies extensively in teaching courses and topics in computing ethics, software project planning, software requirements analysis, team building, design patterns, and software processes analysis. These cases were helpful in teaching “small-scale” software and computing topics, and students were motivated and seemed to enjoy this type of learning activity. However, the case studies addressed issues ranging over a disparate set of problem domains, software engineering practices, and scenario elements; this resulted in students learning about software engineering practices in bits and pieces - there was very little coupling between the case studies and hence no accumulation of scenario experience that allowed progress toward more substantial and complex problems.

The SRS Inspection Case Module (and variations of it) has been used in a variety of courses and workshops: a sophomore-level introductory course in software engineering; an undergraduate course in software quality assurance; an undergraduate course in software requirements; a graduate course in software architecture; a faculty workshop in software process; and a short course in software reliability.

In fall 2009 and spring 2010 the SRS Inspection Case Module was used in three software engineering classes (two sophomore level classes and one senior level class), in which nine inspection teams were formed to carry out the module exercise. The inspections concentrated on the functional requirement statements and used the Inspection Package described in Table 1: Inspection Process Description, SRS Checklist, Defect Log, and Inspection Summary Report Form. The Report Form includes information summary data about the inspections; it is based on the work of Fagan [3] and Humphrey [6].

Table 4 illustrates the results from one of the student inspection teams. The results here are typical (although there were some wide variations in the number of defects identified and time spent on the inspection) and illustrate the types of data collected and the degree to which a teacher can judge the effectiveness of an inspection exercise

In the comments section of Table 4 we compare the team results with benchmark goals for inspection data (based on work by Humphrey [6]). Although the defect removal rate and the overall inspection rate seem reasonable, the team identified only 10 major defects, while DH project team had previously reviewed the SRS and identified over 20 major defects (many purposely seeded in the SRS). Two types of defects commonly missed were compound statements containing multiple requirements and the testability problems with some requirements statements. One conclusion might be the inspection team was ineffective; however, we viewed this more as an education exercise, not strictly a quality assurance activity. By engaging students in “closely” reading the SRS, we helped them to understand its meaning, to determine the degree to which it addressed the customer need statement, to evaluate the correctness, clarity and precision of the requirements statements, and to identify missing features. We believe this sort of “reading” is a critical, first step in helping students (or professionals) prepare for requirements analysis and specification tasks.

The experiences of inspection teams also provided valuable input to the DH project team of how well the SRS Inspection CASE Module worked. This motivated several changes to the module: revision of the inspection process and the SRS Checklist to improve clarity and completeness,

and addition of a comment in the Teaching Notes in Table 2 about the need to emphasize the importance of properly collecting and recorded inspection data.

Table4. Inspection Class Data

Inspection Features	Team Data	Comment
Requirements Size	7 pages	The team only reviewed a portion of the SRS.
Major Defects Identified	10	A major defect either changes the program source code or would ultimately cause change in the program source code.
Major Defects Missed	2	Total defects of 12 were estimated using the “capture-recapture” method [6]
Total Inspection Time	14.2 hrs	9.2 hrs for preparation time and inspections time; one hr inspection meeting (times 5 people).
Defect Removal Rate	0.7 def/hr	A benchmark goal is 0.5 def/hr [6]
Inspection Rate	0.49 pg/hr	A benchmark goal is < 2 pg/hr [6]

6. Summary and Future Work

The use of case study-base teaching has shown success in disciplines such as medicine and law. The DigitalHome case study project’s goal is to enhance the use of case studies throughout a computing curriculum. This is done through the development of case study material that can be tailored to multiple courses and teaching techniques. The case study materials include case modules for teaching software development topics such as requirements analysis, inspections, OO design and construction, and testing among others. Case modules include teaching notes to guide instructors in teaching the target topic. Case modules are also accompanied by class exercises that employ scenarios to simulate real world environment and increase students’ interest in the topics.

As mentioned above, we are continuing to develop the complete set of software artifacts for the DigitalHome system. Beside the material discussed in Table1, we currently have initial versions of the architecture document as well as the system test plan. We plan on having the complete set of artifacts, case modules and related exercises by the end of the project at the end of 2011

As part of the development of the case study material, as well as the dissemination of the results we plan on organizing a summer faculty workshop (in June 2011) to involve other faculty in the development of different case study modules, and use of the project materials at their institutes. Selected participants will be asked to identify a course that they plan to integrate the case study material. In addition, they will be asked to collect some baseline data on learning outcome attainment that will be used as part of the project assessment. In the first part of the workshop, after a discussion of the case study-method of teaching, faculty will assess the portions of the project case study developed to that point. In particular, they will be asked to review the artifacts, scenarios and other support material, and suggest changes and improvements. In the second part of the workshop, the participants will concentrate on the development of case studies that they plan to incorporate in their curriculum at their own university

7. 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

- [1] Davis, C. and Wilcock, E., “Teaching Materials Using Case Studies”, UK Centre for Materials Education, <http://www.materials.ac.uk/guides/casestudies.asp>, accessed November 2010.
- [2] Tomey, Ann M. “Learning with Cases”, Journal Of Continuing Education In Nursing, Vol 34, No 1, January/February 2003.
- [3] Fagan, M, E “Design and Code Inspections to Reduce Errors in Program Development”, IBM Systems Journal, Vol 15, No 3, 1976, pp. 258-287, pp. 744-751.
- [4] Fritz H. Grupe and Joelle K. Jay, “Incremental Cases”, College Teaching, September 22, 2000, pp. 123-128.
- [5] Herreid, C. F. “Case Studies in Science: A Novel Method of Science Education”, Journal of College Science Teaching, February 1994, pp. 221-229.
- [6] Watts S. Humphrey, Introduction to the Team Software Process, Addison-Wesley, 2000.
- [7] Hilburn, T., Towhidnejad, M., “A Case for Software Engineering” Proceedings of the 20th Conference on Software Engineering Education and Training, 2007
- [8] Hilburn, T., Towhidnejad, M., Salamah, S., “The DigitalHome Case Study Material” Proceedings of the 21st Conference on Software Engineering Education and Training. 2008
- [9] Linn, M. , Clancy M., “The case for case studies of programming problems”, Communications of the ACM, v.35 n.3, p.121-132, March 1992
- [10] Lethbridge T, C., Laganière, R., “Object-Oriented Software Engineering: Practical Software Development using UML and Java” Second Edition, McGraw Hill, 2001
- [11] Metseker, S., “Design Patterns in Java”, Second Eddition, 2006
- [12] National Center for Case Study Teaching in Science: <http://ublib.buffalo.edu/libraries/projects/cases/case.html>
- [13] Pfleeger, S, L., “Software Engineering” Pearson, 2005
- [14] Parrish, A., Hale, D., Dixson, B., Hale, J., “A Case Study Approach to Teaching Component Based Software Engineering in Proceedings of the Thirteenth Conference on Software Engineering Education and Training. 2000
- [15] Pressman, R, S., “Software Engineering: A Practitioner’s Approach”, McGraw-Hill, 2007