Case Studies as Assessment Tools in Software Engineering Classrooms

by

Kirti Garg, Vasudeva Varma

in

22nd IEEE-CS International Conference of Software Engineering Education and Training (CSEET'09), Feb 2009, Hyderabad, India

Report No: IIIT/TR/2009/84

Centre for Software Engineering Research Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
May 2009
Case Studies as Assessment Tools in Software Engineering Classrooms

Abstract

Software Engineering (SE) courses intend to prepare students for their professional responsibilities as Software Engineers. A major part of SE jobs involves solving of authentic and complex problems by applying varied SE knowledge and skills as well as skills such as problem solving, critical thinking, use of various tools, communication skills. Typically SE courses are offered as one or two semester course as part of a Computer Science Curriculum. Thus SE courses can have multiple and complex learning goals. Assessment tools and methods such as multiple choice questions, essay type questions, etc. appear to be obvious choices of assessment due to ease in administering, but are not always an optimal choice for such a SE course with large number of learning goals.

Case studies in Software Engineering have been used as research and investigation tools since long, and as teaching tool recently, but not as assessment tools. In this paper, we describe and demonstrate the use of case studies in SE as assessment tools in Software Engineering classes. We developed a set of parameters that are derived from the course learning goals and help in objective as well as easy evaluation of case studies solutions. This work reasons that carefully designed case studies can be effective as well as efficient assessment tools for Software Engineering education such that they are not only closely aligned with learning goals, but consist of many characteristics of an ideal assessment tool in accordance with Learning Sciences.

1. Introduction

Assessments have always been a source of worry for course designers. The problem becomes even more troublesome for a course like Software Engineering that has a large number of learning goals that too from higher cognitive levels. Further difficulties lie in creating meaningful assessments that satisfy the Learning sciences. Good assessments are aligned with the course learning goals, integrated with the instruction and are not a burden on the faculty and students. Further the assessment should provide feedback as well enhance learning opportunities to students [13]. A bigger role that assessment in an SE course should play is to make the learning competence oriented and not performance oriented [1][7], as this competence would be shaping a student’s career as a Software Engineer.

Since assessment and learning objectives should be in synchronization with each other, hence it is important to examine the nature of learning goals in SE education. Usually SE educational endeavors have multiple and complex educational goals. These consist of usual knowledge and understanding components of SE factual, conceptual, procedural and meta-cognitive knowledge (referred as SE knowledge further in this paper), related skills, application of these knowledge and skills, soft skills such as problem solving, communication, team-building etc. It is important to note here that the problem solving skills include higher order cognitive skills such as analysis, synthesis and evaluation [2]. Understanding and analysis skills are important in order to get well versed with the problem space. Though multiple and complex, these are essential goals because whole purpose of the course is that a student should get well versed with the required knowledge and skills and should be able to apply them in order to develop software solutions for complex real world problems.

Many a times, students are enrolled in a Computer Science program that has SE as just one of the offered courses. In such situations, students get only one chance to learn Software
Engineering before they start working as Software Engineers. It is important that this one semester course equips the student with essential knowledge and skills required by a Software engineering professional.

Considering the nature of SE education, the assessment method being used will be considered effective only if it takes care of at least the following:

a) Should cover the breadth as well depth specified by learning goals of a course
b) Should evaluate the transfer of knowledge and skills under realistic scenarios i.e. application of knowledge and skills for authentic real world problems
c) Should consider the time and resource constraints on the course
d) Should assess deeper learning

Course designers often find it difficult to precisely define such complex course learning goals and pick or design an instruction approach that caters to all the learning goals effectively, and can be realized within limitations of time, resources, and experience.

This paper records our first impressions and experience of case studies as assessment tools for SE education. We will discuss various issues associated with assessments in SE education. Then we will reason about the use of case studies for assessment in SE education. Solving carefully crafted case studies requires not only a firm understanding of SE knowledge (conceptual, procedural and meta-cognitive), but also application of this knowledge and skills to solve authentic and complex problems. Solving a case study provides opportunities for students to demonstrate their competence in problem solving, analysis and synthesis skills as well since case studies will require a thorough understanding and analysis of the problem, designing reasonable solutions and all in the light of SE knowledge. Thus, there use is well aligned with learning goals of SE education.

We will discuss an evaluation framework for case studies that helped us to reduce the subjectivity from case study solution, bring in objectiveness and still provide sufficient scope for creativity. We explain this approach through our experience of using case studies in a Se course. Further we discuss how case studies satisfy the expected properties from an ideal assessment tool from a learning sciences perspective. We also discuss some of the factors to be considered while using case studies for assessment. Please be cautioned that these are our first impressions. Though we arrived at the logical conclusion that case study are potential candidates as assessment tool in SE education, further empirical investigations are a must to judge further reliability and validity of case studies for assessment.

2. Related Work

Not much literature is available on designing assessments for Software Engineering education courses. Jody Paul discusses use of hypermedia based non traditional assessment tools for SE courses [7]. A more recent work by Golden and Bass gives suggestions for development of meaningful assessments in software architecture courses for software professionals [6]. In a previous paper, Jody discusses how assessments can be vehicles of learning in [8]. Jody also shares the view with Resnik et al in [11] and Priestly [10] that traditional assessment methods are not sufficient for assessing higher order cognitive skills. Use of stories (in form of case studies, narrations or scenarios) for instruction has been advised by Dalcher [3], Garg and Varma in [5] and many others [14],[15],[16].

3. Common Assessment Methods

Most of the SE courses rely upon traditional assessment methods such as subjective questions, long essay type questions, multiple choice questions (MCQs) or project deliverables etc. Our experience with SE course assessments in that faculty tends to use
traditional assessment tools because as they are easy to develop and use, and are being used traditionally. Many issues arise out of this use of traditional assessment methods. We will discuss the most important ones here.

Presence of higher order cognitive learning goals add to difficulties in creating a meaningful and useful assessment that can cater to all learning goals. The traditional assessment methods such as MCQs and essay type questions are suitable for the knowledge and understanding learning goals, but not much useful for other goals such as application, analysis, synthesis and evaluation [10][11][1]. This is true for education in all disciplines. For a SE course, this means that use of only the traditional assessments violates the basic learning science principle that learning goals, instruction and assessment should be well aligned [9] and assessment should cover all learning goals.

Some courses are project centered where projects are used help students apply knowledge and skills. But these projects usually represent a very highly abstracted view of reality to honor time and resource constraints (Thus called as academic projects). Sometimes students focus on the project outcome (product) and not the process. Hence, by working on these projects, students may not experience the actual complexities of software development. Thus competence of a student in applying skills or synthesis of learning in real world scenarios should not be established for students from an academic project based assessment.

Traditional assessments methods make the learning to be performance centric. Students will spend there time and energies on knowledge and skills that hold the promise of better grades. Since traditional assessments rarely include exercising of higher order cognitive skills, students avoid learning and practicing these crucial skills.

We miss on many advantages that carefully designed assessments offer to a course. We will discuss these advantages when discussing characteristics of case study based assessment.

4. Case Studies for Assessment

In this section we will describe the use of case studies for assessment. First we will describe the basic of the approach. Then we describe a unique framework (grading rubric) to evaluate case study solutions. It not only serves a large number of learning goals, but also reduces the subjectivity from case study based assessment. We describe the usage of this approach in one of our SE courses. Next we describe how the case study approach satisfies various characteristics of ideal assessments as given by Learning Sciences. We conclude this section by describing various factors to be considered while using case studies.

4.1 Basic Details

Bromley defined case study as a "systematic inquiry into an event or a set of related events which aims to describe and explain the phenomenon of interest" [1].

We define a SE case study as an account of a Software Engineering (development) activity, event or problem containing background and complexities actually encountered by a software engineer.

In order to use a case study as an assessment tool we augment some challenges or problems to this case account or narration. The challenges/problems need to be ‘solved’. Some of the possible forms of challenges for a given scenario in software development are:

- Find a promising solution and justify it using SE principles
- Given two or more alternatives, pick the most suitable one
• Analyze and describe why a particular problem occurred in first place (what went wrong)
• Given outline of a solution, implement the solution
• Use a particular tool to achieve the given purpose
• Identify good and bad practices as observed. Justify using SE concepts

Solving such challenges requires wide variety of knowledge and skills. It requires application of some knowledge and skills to understand and analyze the situation, come up with a solution (designing alternatives, evaluating alternatives, decision making) and finally application of knowledge and skills to implement the solution. In fact, even when a SE course explicitly does not have the higher order cognitive skills (analysis, synthesis, evaluation) as part of teaching goals, then also the ‘case solving’ helps to inculcate these skills subtly. Application of skills would usually yield some deliverable such as a design document, requirement specification, test case etc. that can be further evaluated.

Students may be asked to present and discuss their solutions in class and as written reports. These deliverables become the ‘observations’ for assessment and can be objectively graded by the faculty as well as can be shared among peer students for feedback. The reporting of solutions can be used to hone as well as assess communication skills. The final grade may be based on case solution (as a report or otherwise) and class discussion. A triangulation assessment tools can be used for fairer assessments.

4.2 Evaluation of Case Study Solutions

A major challenge in the case study based assessment is to devise the schema (rubric) to evaluate the case study solutions. This rubric should make the evaluation as objective as possible and should be easily generalized for different courses and different types of case studies. Further, the schema should be closely linked to the learning objectives of the course. We use two basic approaches to satisfy these requirements: A) Use the SE course learning goals to derive a set of parameters for objective assessment. Whatever are the learning goals, they can always be reduced to one of the six cognitive goals of education [2]. Soft-skills such as communication and team building can be listed as separate categories. Now for each of the six categories, we developed a set of scoring parameters that are qualitative, and can be converted into quantitative terms if needed. Table 1 lists the broad level learning (cognitive + other) goals, and their evaluation parameters.

Since the challenges in the case study are also based on the one or the other cognitive goals, hence if we evaluate their solutions as per the pre-decided parameters, we should be able to draw conclusions about the competence of a student for those particular goals.

B) Any solution given by the student is valid if it is based upon Software Engineering concepts and is justified. Thus there is not a single right answer, though there are certain wrong answers. All solutions given by students are accepted, provided they give a full justification using SE knowledge. A student loses points only when the piece of knowledge being used or stated is incorrect or is not applicable under given circumstances or the analysis of the case scenario was incorrect.

<table>
<thead>
<tr>
<th>Learning Goal</th>
<th>Scoring Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge</td>
<td>Can state or identify Software engineering principles,</td>
</tr>
</tbody>
</table>
Understand | Can summarize the case study correctly, Restate SE knowledge in own words,
---|---
Application | Can state all pre-conditions and post-conditions related to application of SE knowledge and skills, Successful and complete application of SE procedure (skill) or tool to produce the specified deliverables
Analyse | Can identify patterns, problems, issues, good and bad practices Differentiate root cause from symptom of a problem/issue, Identify components of a software system, Explain how patterns/ideas connect or do not connect to each other
Synthesis | Can state the relation between two different things, Refers to previous discussions/learning, Combine ideas to produce an explanation, Justify the application of some knowledge/skill
Evaluation | Can state $n$ number of correct and most important pros and cons (from SE point) of two different proposals
Communication | Clarity of presentation, Use of example

Use of case studies for assessment has always been subjected to skepticism due to fears of subjectivity while evaluating case study solutions. By using these two basic principles we reduce the subjectivity to a large extend, though we do not claim its complete removal.

Problem solving deeply resembles the software development process and hence solving such problems also helps us to build meta-cognitive skills as well.

4.3. An Example

We offered a first course in Software Engineering to Computer Science undergraduates and graduates at our university. MCQs were also used as assessment tools for knowledge component. Case Studies were being used for both instruction and assessment in the course. During instruction, the case solving exercise was done in groups and solution reports were used for continuous assessment.

Case solving was used independently for assessment during the end-semester examination. We discussed the idea of using case study for assessment in class before using it for first time. Initially, students were apprehensive that this will require more preparation and efforts on their part. We gave them choices of two cases and posted the case 3 days in advance on course website. Students were informed that one of the two cases would be randomly picked and asked in the exam. The cases were given in advance so that student are familiar with the context and need not spend time in understanding the case during the limited exam time.

Case Study A was about “Requirements engineering and development model for a Student Information System for University”. Students were already familiar with this system. The challenges involved correctly and completely stating different types of requirements (functional, non-functional, user, system etc.) for the system. They were also asked to draw use cases, interaction diagrams and develop a test suite (test cases, test data) for two of the most important functional requirements. Another challenge was to list the architectural drivers. Finally, given stringent constraints on development time and resources, students were asked to select a suitable SDLC model for the project and justify the choice.

Case Study B was based on Integration of internal applications of an enterprise. These applications included Time-sheets, employee databases, leave management, invoice management etc. Now the applications should be integrate to avoid redundancy and ease communication between them. A set of stringent non-functional requirements were given that included scalability, role-based access, security, performance, single sign-on etc. The challenge was to design an application that works as a single portal for all these applications with uniform look and feel. For this students should restate the requirements so that they are
non-ambiguous and complete, come up with a high-level architecture, draw use case
diagrams, and logical diagram. They should also pick a suitable SDLC model for
development of this component based system and justify the choice.

A summary of learning goals from two cases is as follows:

a) Knowledge Aspects: definitions of different types of requirements and architectural
drivers, procedure of stating requirements in complete and meaningful manner.

b) Skills: classification, disambiguation and completion of requirements, UML

c) Analysis: of requirements to identify different types of requirements from given case
description, identification of different components of the given systems

d) Evaluation and synthesis skills: to pick a suitable SDLC model and justifying the choice,
justify the choice of most important requirements.

students reported that they did not find the case solving exercise to be burdensome.
Students also reported that though initially they feared that preparing for exam will take a lot
of time, but they could focus on understanding the concepts instead of preparing for standard
questions and answers from end of the textbook chapters. A couple of students mentioned
that they learned something new while working on the case solution. Most students
complained that working on case solution was very time consuming.

We did not pre-decide a model solution. We used a scoring ruberic, where we assigned
points to each challenge. Each student was given full points initially, and a fixed number of
points were deducted when the student committed a mistake (no justification, incomplete or
incorrect diagram, incomplete or wrong classification of requirements, ambiguous
requirements, etc). For example, consider the evaluation question “Suggest which SDLC
model would be most suitable”. Every student who listed the SE related pros and cons of
selecting two models for the given situation, selected one of these two models and justified
the decision by using the pro and con list scored high for this challenge.

Students could justify their choices and did well on understanding, application and analysis
skills. We found that students came up with acceptable quality solutions for the case and
performed better than expected.

4.4. Accordance with Learning Sciences and Advantages

We will discuss some of the major advantages of case studies in light of the characteristics of
assessments as given by learning sciences [9][13].

4.4.1. Alignment of Goals and Assessment: It is clearly evident that the assessment is
well aligned with the goals since we have the ability to measure all the goals including all
knowledge, skills and even dispositions, soft-skills and problem solving under real-world
situations. Problem solving deeply resembles the software development process and
hence solving problems will help to build meta-cognitive skills as well.

4.4.2. Motivation towards competence building: The use of case studies for assessment
would help even the performance oriented student to build competence. We discussed earlier
the disadvantage of performance centered assessment. As scoring in the proposed assessment
approach directly related to competence in all cognitive dimensions, it would encourage
students to give due attention to all aspects of learning. Also, contextualization by help of
real-world complex problems motivates the students to learn further [9].

4.4.3. Provides Learning Opportunities: Students reinforce the knowledge and skills as
they apply it while solving the case. Each case study exercise makes the students aware of
domain and the real world problems faced by Software Engineers [5][3][11]. When students
discuss the solutions, they learn about many more possible solutions and learn analytical
examination. Presentation, discussion, and reporting of case solution hones the
communication skills of students. Case studies help to make connections between the problem patterns and knowledge, by showing how the knowledge can be applied under various contexts. Thus, this would help to build expertise in the subject [1].

4.4.4. Embedded into Instruction: Case studies have been used as effectively learning/teaching tools for SE education [11][3][14][15][16]. Since case studies provide opportunities for learning, these opportunities can be utilized and made a systematic part of the instruction. In such a case, the assessment would not be a separate activity, instruction will include solving of cases and the artifacts produced during instruction can be used for assessment. This will reduce the time and effort requirements for assessment and will not be extra burden of faculty and students. Also, the assessment would become continuous as compared to when done as separate activity, another desired aspect of an assessment [9].

4.4.5. Feedback Opportunities: When students discuss (examine, critically analyze, or appreciate) solutions in class, they get instant feedback from peers and the faculty. The faculty also gets a feedback on student’s learning.

4.5 Success Factors

When using case studies in our class room, we noticed that following factors affect the effectiveness of case studies for assessment in SE courses.

An important issue associated with use of case studies is creation of appropriate case studies. The case should be small enough to be solved in limited time and should provide all necessary data that is required for application of knowledge and skills and to draw meaningful conclusions. Most SE teachers should be able to take existing projects and present them in a narration format, while building appropriate issues and challenges into them.

Full potential of this assessment approach can be realized only when all the involved parties i.e. students and faculty are clear of its purpose and the approach. The students must be aware of case solving methodology. Teachers should form challenges that reflect there specific learning goals and can discuss them with students at the end of the course.

Educators can get skeptic that case studies are useful in small classrooms only. We recommend a grouping pattern to address this issue. Form groups of students and ask the groups to divide the work among group members. Each group member can be asked to present their part during the class discussion. Any member could be asked questions by their classmates during the discussion. Groups should submit a report that clearly states efforts of each member. Triangulation of assessment methods can be used for correct evaluation of every student. We tried this pattern for a class of 124 and it worked well.

Though the evaluation of case solutions may take some time initially, but we experienced that after some practice, the evaluation time was little higher than essay type questions.

A triangulation of multiple assessment tools can be used for concrete evaluation.

5. Conclusion

In this paper, we discussed the need of proper assessments that cater to complex and large number of learning goals in Software Engineering classrooms. We proposed that carefully designed case studies are effective and efficient assessment tools in Software Engineering educational endeavors. We could establish the logical argument that case studies are suitable for assessment as they not only help in evaluating a student’s competence on important, complex and multiple learning goals of SE courses, but can be aligned with learning goals, can be embedded with instruction, provide opportunities for feedback and learning and do
motivate students. We could also remove subjectivity from assessment to some extent and could use them successfully in large classes.

In future we want to devise generic scoring rubrics which can be used for more objective and transparent use of case studies for assessment. Further we want to conduct action research and controlled experiments to check the reliability of case studies for assessment in SE courses.

Though these are our first impressions about case studies as assessment tools, we have a good starting point and can confidently claim that carefully crafted case studies exhibit great potential as assessment tools.

6. References


