Advanced Software Design: a Case in Problem-based Learning

Jocelyn Armarego
School of Engineering,
Murdoch University, Western Australia
jocelyn@eng.murdoch.edu.au

Abstract
The restructuring of a course in Advanced Software Design to address issues raised by students wishing to engage in more ‘real-world’ scenarios provided the opportunity to develop a problem-based learning environment. This paper focuses on the requirements of such an environment and discusses initial feedback on its application.

1. Introduction

1.1. The wickedness of software development

The development of software systems is described in the literature as a wicked problem, characterised, as well as by the lack of precise problem definition, the difficulty of verifying and validating proposed solutions and the inadequacy of methods and methodologies in guaranteeing the quality of the result [1, 2], by:
- the essential uniqueness of each system
- solutions are good/bad, not true/false, which means the designer has no right to be wrong
- any discrepancy in representing a wicked problem can be explained in a variety of ways; the choice of explanation determines how the problem will be resolved [3].

To add to the problems in developing software systems, the inadequacy of formal education in training competent analysts/designers has also been noted [4], although perhaps not satisfactorily explained.

The educational dilemma in teaching Software Engineering is that some tasks may also be classed as wicked problems, where:
- complexity is added rather than reduced with increased understanding of the problem
- metacognitive strategies are fundamental to the process
- a rich background of knowledge and intuition is needed for effective problem-solving
- a breadth of experience is necessary so that similarities and differences with past strategies are used to deal with new situations.

Requirements Engineering has been described as wicked [5], as has Software Design and system Testing [6]. The curriculum addressing these components requires a foundation in content to be balanced with elements of creativity and experience based on practice.

1.2. Learning for wicked problems

These requirements may best be addressed through less traditional approaches to learning, with a focus on advanced knowledge acquisition (as the stage between novice and expertise
As one example of an alternate approach, problem-based learning (PBL) integrates the learning of content and skills in a collaborative environment, and emphasises "learning to learn" by placing great responsibility for learning on the learner [8]. Acknowledging both the ill-structure [9] and multidisciplinary [12] nature of software development, it has been argued that:

- learning based around constructivist principles is likely to be more suitable in domains involving ill-structured problems [7]. These principles are encapsulated almost ideally in problem-based learning [10]
- appropriate learning in ill-structured domains and/or dealing with ill-structured problems should itself be problem-based
- problem-based learning best provides an effective environment for future professionals who need to access knowledge across a range of disciplines [11].

This learning model has been applied to numerous areas of higher education (social work, architecture, law and engineering), although it is best documented with medicine.

As an ideology, problem-based learning is rooted in the experiential learning tradition, but with a number of different forms according to the nature of the field and goals of the learning situation [11]. Through its emphasis on problem and student-centredness, PBL is seen to [13]:

- acknowledge the base of student experience
- emphasise student responsibility for learning
- cross boundaries between disciplines
- intertwine theory with practice
- focus on the process of knowledge acquisition rather than the products of that process
- change staff roles from instructor to facilitator
- focus on student self and peer assessment
- focus on communication and interpersonal skills so that students understand that to relate their knowledge, skills beyond their area of technical expertise are required.

It has been argued [14] that PBL is an educational strategy that required three components to be differentiated:

- an integrated curriculum organised around real-world problems rather than disciplines and with an emphasis on cognitive skills
- small groups, tutorial instruction and active learning conditions to facilitate PBL
- outcomes such as the development of skills and motivation together with the development of an ability to be lifelong learners.

A focus on authentic problems as a context for learning also accords well with theories of expertise - learning beyond the initial stages may best be achieved through situational case studies with rich contextual information [15]. Its supporters claim PBL results in increased motivation for learning, better integration of knowledge across disciplines and greater commitment to continued professional learning [11]. As well as offering the flexibility to cater for a variety of learning styles, the focus moves to dealing with information in ways that reflect how learners might use it in real life [16]. These characteristics suggest that PBL is worth investigating for the solving of wicked problems in wicked domains.

2. Background to the case-study

Advanced Software Design II (ASD II) is currently a course offered in the final semester of a 4-year undergraduate engineering program in Software Engineering (SE). It comprises the last of the eight core SE courses on offer, and is undertaken concurrently with an (industry-
based where possible) individual project and a course in Engineering Law. The impetus for the restructure of this course is 2-fold:

- final year students have raised issues regarding the need to apply knowledge they have acquired to more ‘real world’ scenarios – and to engage with the material on offer
- a restructure of the program for 2002 will mean a reduced opportunity for group-based projects. While the current intake of students has participated in a course centred on a university-based small group project, this will not necessarily be the case from 2002.

3. Applying a PBL methodology

One taxonomy [17] proposes several varieties of PBL in use. These describe a continuum, from lecture-based cases where case material is used to demonstrate lecture information to problem-based and closed-loop problem-based (students meet with a client in a simulated environment that allows free enquiry). The type of scenarios offered, the assessment methods, learner autonomy and the way in which teaching and learning occur determine which variation is most appropriate for any given environment. All varieties, however, should conform to a problem-based learning methodology [18]. This incorporates five recursive stages:

- problem analysis - the rich context is mined for important facts, sub-problem(s) and alternate solution paths generated
- self-directed learning - the learning agenda is determined by the information needed to evaluate the alternatives proposed
- problem re-examination - based on findings, solution paths are added, deleted or revised
- abstraction - an articulation process to increase the utility of the knowledge gained in specific contexts
- reflection - a debriefing of the experience to identify improvement in the learning process.

3.1. Problem analysis

The School of Engineering is fortunate in incorporating a Pilot Plant that simulates the Bayer process for mineral extraction. The students were required to design a software simulation for this process, based on the configuration in the Pilot Plant. Prior to the students encountering any course material (other than mandatory course outlines and assessment guidelines) the problem was presented and associated documents made available. These provided the context for the problem – the components (eg tanks, valves, pumps etc), their attributes and constraints. All students were marginally familiar with the workings of the Plant – other courses within their program had made use of the facility. In addition, as well as the lecturer as facilitator for the project, a student undertaking a PhD based on the Pilot Plant (and specifically working on an object model of it) agreed to act as expert client. All preliminary documentation had been prepared in consultation with this student, so a consistency of approach was reasonably assured.

At this point (the first course session – week 1) students were asked to raise any issues they could see causing problems. Discussion centred around

- team size
- quality of the product at each phase
- dependence on peers for components of the final grade.

These are discussed below.
3.2. Self-directed learning

While the problem drew on knowledge acquired throughout the SE program, the broad focus of the ASD II course is on software architecture. Course material (as a set of introductions and readings on architectural issues) was made available on-line. Students were also given access to the online material for all the SE courses they had previously completed - to provide them with resources to streamline the process of locating necessary information within a PBL framework [19]. It is important to note that drawing only on previous knowledge could not solve the problem set: architectural and other issues had to be explored and applied (so for example, Object-Z was introduced, although the students were nominally comfortable with Z, and an understanding of design patterns (new material) was required to solve a major component of the problem). By the same token, students could not solve the problem by ignoring previous knowledge (eg in SE process, tools (eg use of UML and Rational Rose was mandatory), team/project management).

The 13 students enrolled in the course were randomly assigned to groups based on the (physical) components of the Pilot Plant (so, for example, one group was required to investigate and model the properties of the various types of tanks required by the system, another the valves, etc). These groups were required to present a composite model for each phase of the problem solution. After each submission the groups were dissolved and reformed, again randomly, but ensuring no student worked on the same component in two consecutive sessions. Each group was also required to nominate a liaison person (for interaction with other groups) - these liaisons became the management team.

Classes were conducted as a 2-hour workshop each week – the first half a summary and discussion of an architectural issue, the second a forum managed by the students (generally the liaison group) for discussion and interpretation of the project. Students were expected to complete another 10-11 hours outside class time each week (although this is optimistic).

After the initial 3 weeks of the semester, students conducted both the architectural issues section of the workshop as well as the project forum. Very quickly (by week 5) the issues component of the workshop became tailored to the needs of the project – how the knowledge being discussed could be applied to this specific problem.

3.3. Problem re-examination

Students are required to provide a phased solution to the problem presented:
- Requirements model – due week 5
- Architectural model – due week 7
- Low level Design model – due week 11.

The first model developed, the Requirements Model, may be considered the end of the initial problem analysis stage of the PBL methodology. At this point the problem is generally well understood, and a set of approaches (in this case assuming object technology) determined. Once groups were reallocated, the primary task of each group was to evaluate the model submitted based on the needs of the next solution phase. A marking scheme was provided for each group’s use. The final mark for each assessment component was based on the average of the peer assessment and the teacher mark. Interestingly, anecdotal evidence within the School shows that peer-assessment generally results in a lower mark than that allocated by the teacher.

Based on the team-based critiques provided, the model was modified (and modifications justified) in order to produce the next phase output. The ‘new’ group undertook these modifications, often with input from members of the original team. This achieved several aims:
deficiencies were corrected through a process of negotiation between original and new teams

- the model was not modified simply to make the task ‘easier’ – a sense of ownership on the part of the original team minimised this
- the need to explain/justify approaches taken fed into future stages of the PBL methodology, which focus on articulation and reflection.

3.4. Abstraction

The process of learning involves traversing the granularity of various disciplines to varying extents, from detailed to abstract and from intrinsically simple to complex representations of knowledge [20]. In addition, it is suggested students reason at many grain sizes, especially in reference to problem solving abilities - as they refine their understanding, students articulate their knowledge to finer grain size along the dimensions of:

- aggregation - progression from part to whole integration in a framework
- abstraction - higher order concepts and relevance-based simplification
- goals.

Students also move in the opposite direction - from fine-grained knowledge of particular situations to an understanding of inclusive, generic, coarse-grained knowledge [20].

As has been noted, course content was described and discussed in terms of the problem requirements. What is also required is the ability to generalise from the specifics learnt. As one student put it in week 5 I wish I had known about some of this before I started my thesis ¹. As the course progressed, comments such as this became common place.

Articulation of the knowledge gained took several forms:

- student-led description/discussion of a specific software architecture topic. As noted above, students very quickly tailored the session to the needs of the project
- the models produced needed to be accompanied by documentation detailing the decisions made and their rationale. This provides support for the evaluation undertaken
- assessment required a critique (with rationale) of deficiencies as well as the merits of the model produced (basically addressing how could have been better done, and why).

3.5. Reflection

Competent problem solving behaviour includes the ability to monitor and assess what one does when working problems, using all resources available [22].

Reflection occurred both on the macro and micro scale. At each workshop session, students were given the opportunity to comment/reflect on the learning environment. This was especially evident in the first 4 weeks of the semester, where the group dynamics were being tested and both peer and student/teacher interaction determined. The initial reaction to the open nature of the learning process was somewhat negative (for example see the comments at the initial School-based course survey). However, as the semester progressed, students both appreciated the flexibility this provided them, and were willing to take control/accept responsibility for the learning they were undertaking (a week 5 comment “this is magic!”).

¹ The re organisation of the BE(SE) program from 2002 ensures this will be the case. ASD II moves to second semester 3rd year, before the thesis is commenced.
Reflection at a more macro level also took place – discussions about individual learning styles, the merit of verbal/visual, global/sequential learning occurred, in the context of how this learning environment could be improved (at both the individual student and course level).

4. Issues raised and addressed

From the outset of the course, students were given the opportunity to raise and discuss issues (as was the teacher!).

Student-based issues included:

➢ the majority had not worked in such a large team previously – their experience of team work was confined to groups of three or four working independently of other groups. This task required the small, component-based teams to integrate their material with that of the other teams to provide a consistent solution to the problem

➢ side issues - management structure, roles and responsibilities and communications between the teams had to be dealt with

➢ delegation of work – a management team based on a liaison person from each team were required to institute standards and procedures to facilitate the integration required. Members of this team soon found the burden of ensuring consistency overwhelming if these were not enforced (as noted by one liaison person in week 5 when you work in small groups you can afford to be democratic as a comment on the amount of effort being expended by the management team to provide a consistent approach to the solution)

➢ subsequent phases were totally based on the output of the previous phase – teams were concerned that the quality of the product would not be ‘up to standard’. All students were required to assess the product of the previous phase – allowing for reflection – how could it be done better, and adaptability – what changes/additions need to be incorporated in the next phase to neutralise deficiencies detected. This review process and the peer-assessment grade that was its outcome were added to the lecturer’s assessment to become a component of the final grades awarded

➢ side issue - reliance not only on other members of the team, but on members of other teams to produce acceptable results. A high level of collaborative interaction was required for the problem to be solved successfully.

Teacher-based issues related to administration and evaluation of the experience, and are described below.

4.1. Administration

Within an environment that could provides opportunity for students to value their own perspectives in the learning process and to argue their interpretations of the problem and its solutions, an operational curriculum is in fact bounded by the context in which it is placed. In particular, the teacher’s pedagogical stance influences the way problem-based learning is enacted in practice [13]. It therefore becomes a challenge for the teacher to become

an orchestrator of opportunities for learning (in its widest sense)[13]

This quote is taken from a series of models that illustrate how learners are enabled/disabled in the process of knowledge construction within a PBL environment. The underlying aim of the restructure of the ASD II course has been to move from a lower-level model to one providing a richer PBL environment for the learners. These models are briefly described below.
Table 1. Model II PBL for professional action

<table>
<thead>
<tr>
<th>knowledge</th>
<th>practical and performative</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning</td>
<td>outcome focussed acquisition of skills and knowledge for the workplace</td>
</tr>
<tr>
<td>problem scenario</td>
<td>focussed on real-life situations that require an effective practical solution</td>
</tr>
<tr>
<td>students</td>
<td>pragmatists inducted into professional cultures who can undertake practical action</td>
</tr>
<tr>
<td>facilitator</td>
<td>a demonstrator of skills and guide to best practice</td>
</tr>
<tr>
<td>assessment</td>
<td>testing of skills and competencies for the work place supported by a body of knowledge</td>
</tr>
</tbody>
</table>

Model II (Table 1) addresses professional action and focuses on know-how, which will allow students to gain competence to practice within given frameworks and is seen to apply within curricula that have strong links with industry and are influenced by the world. Students are expected to transfer skills acquired to the world of work, but without them necessarily being rooted in cognitive content and professional judgement. Within the SE environment, this is seen as a deficiency of the model.

Model III (Table 2) allows problem-based learning to bridge the gap between know-how and know-that, with a focus on understanding and synthesising information rather than gaining a particular depth of coverage. The discipline boundaries imposed mean the learner must make the connection between fragments of knowledge:

Table 2. Model III PBL for interdisciplinary understanding

<table>
<thead>
<tr>
<th>knowledge</th>
<th>propositional, practical and performative</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning</td>
<td>the synthesis of skills and knowledge across discipline boundaries</td>
</tr>
<tr>
<td>problem scenario</td>
<td>acquiring knowledge to be able to do, therefore centred around knowledge with action</td>
</tr>
<tr>
<td>students</td>
<td>integrators across boundaries</td>
</tr>
<tr>
<td>facilitator</td>
<td>a co ordinator of knowledge and skills across boundaries of both</td>
</tr>
<tr>
<td>assessment</td>
<td>examination of skills and knowledge in a context that may have been learnt out of context</td>
</tr>
</tbody>
</table>

Neither of these models provides the student with the ability to transcend imposed frameworks, whether those of disciplinary boundaries or of personal stance. Model IV (Table 3) at least allows the student to acknowledge that these boundaries exist.

Reflection on and an openness towards the stance of others implies an evaluation of one’s own [13]. In this model, students are encouraged to develop an autonomous position as individuals within the group, and as a group. Students take a critical position towards knowledge, themselves and their peers and elect to use the group to resolve dilemmas. The challenge on the teacher is to focus on quality of product, and provide feedback to the group, as well as facilitate the process.
The ASD II course aspires to model this PBL environment.

Table 3. Model IV PBL for transdisciplinary learning

<table>
<thead>
<tr>
<th>knowledge</th>
<th>the examining and testing out of given knowledge and frameworks</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning</td>
<td>critical thought and decentring oneself from disciplines in order to understand them</td>
</tr>
<tr>
<td>problem scenario</td>
<td>characterised by resolving and managing dilemmas</td>
</tr>
<tr>
<td>students</td>
<td>independent thinkers who take up a critical stance towards learning</td>
</tr>
<tr>
<td>facilitator</td>
<td>an orchestrator of opportunities for learning (in its widest sense)</td>
</tr>
<tr>
<td>assessment</td>
<td>opportunity to demonstrate an integrated understanding of skills and personal and propositional knowledge across disciplines.</td>
</tr>
</tbody>
</table>

4.2. Feedback

The School of Engineering requires that all students be surveyed at two points within a 13 – week semester, at week 4 and week 11. Students are asked to comment on good and bad elements of each course undertaken, as well as making general comments. These surveys are based on year-groupings and managed by members of academic staff designated as year-coordinators. They are for internal use only and ensure issues are quickly identified, discussed with appropriate course controllers and feedback provided to students. In addition, the University undertakes student assessments of teaching and course material towards the end of semester.

The week 4 survey highlighted the following issues:

- some students were concerned about the seemingly open nature of the problem – they felt there was not enough direction provided on how the problem should be
  - analysed
  - tackled

  Response – enough information was provided in the problem description to answer all the student queries. However, some students were not in a position to assimilate this material as they had not yet spent enough time on the background/context documentation

- management of the project – this was the initial issues recurring, voicing concern on
  - size of the project
  - structure and organisation of an ongoing project

  Response – the problem itself was not in fact very big – students were marginally (at least) familiar with its context and had access to relevant resources. However, they found the idea of a 13-member team initially overwhelming. Encouragement that it could be done, and guidance on management/communications structures alleviated some of their concerns. Access to document templates and standard procedures on a shared area dedicated to the project also provided some of the structure they required

- students concerned that their assessment is based on the motivation and performance of others too many people that depend on each other’s grade
Response – the students required time to get used to the PBL environment. Initially, students felt threatened by learning methods employed. While they were happy to accept responsibility for their own learning and performance outside of a formal course (in particular within the thesis, based on an industry project), their expectations were for formal courses to be teacher-led to a greater extent. This was different from anything they had done before.

As the semester progressed, further feedback was sought. This both helped the teacher fine tune the process and provided the student with a (somewhat) anonymous mechanism to voice any concerns they may not be willing to have openly attributed. Project-specific feedback was requested in week 7 (by which time phase 2 was due, although in fact the schedule had slipped by one week). Students were asked what they would add, change or delete for any aspect of the project as well as asked to comment generally on good and bad points of the course/project structure. Sample comments included:

- add – activity logs/timesheets to feed to project management software (eg MS Project)
- change – maintain original team structure (rather than re-assemble at each stage); Pilot Plant “beyond our comprehension”; more guidance for management group
- delete – [nothing specifically].

Good points overall focussed on the value of the experience and the ‘authenticity’ of the task and may be summarised by the following comment from one student:

*we learn so much ‘practical’ stuff from this project, it would be good to get another chance to actually do it right.*

Bad points focussed on communications, in the form of team guidance (from the teacher) and leadership (from the students):

*no explicit team leadership either by group or individual. Depending on mix of people, this can produce good or bad results*

and

*if the liaison group members are not good at communicating with their teams, then the workload remains with the liaison group and is not equally distributed*

These were generally confirmed in the week 11 survey (semi-colon indicates different students):

- Good: openness of unit, ability to explore concepts; learnt a lot about design skills and approaches for problems; interesting experience (group)...
- Bad: project expectations vague, amount of content; direction, expectations of understanding sometimes unclear, main points hidden amongst lots of information.

4.3. Evaluation

Discussions with employers of our graduates show that course content and grades are no longer a major issue in graduate employment scenarios:

- the program is gaining a reputation for the breadth and quality of the Software Engineering material covered
- high grades do not ensure the graduate will easily integrate into the organisational structure.

The focus is on this latter point – how students transition to being professionals within an industry.
The PBL setting helps and encourages students to practice and develop the skills they would need to act as professional Software Engineers. Based on a framework of generic higher education skills proposed [23], the course could be seen to address:

- **management of self** – the task was seen to be large and unstructured. Students were required to conform to time and task constraints, confront unexpected hurdles and use a variety of learning techniques
- **management of others** – in the group setting, students were required to negotiate and defend their position, while maintaining a good working relationship throughout the semester. Honesty (especially of criticism) was seen as an issue
- **management of information** – interpretation of information provided/sought and catering/coping with multiple perspectives
- **management of task** – both at phase level and overall. Students instigated a project plan for the complete semester, then carried out (and enforced as required) a course of action.

Shortcomings of this application of a PBL environment have been identified. Some have been addressed; others will be incorporated into a review of the course at the end of the year/semester. Prior to that, the evaluation undertaken can only be considered formative – it enables the teacher to be flexible based on student (and teacher) concerns/needs.

**5. Conclusion**

Whatever the nature of the problems or the sequence of learning activities, these common characteristics are shared within a PBL environment:

- the starting problem for learning is a problem which is one the students are apt to face as future professionals
- the knowledge the students are expected to acquire is organised around problems rather than disciplines
- most learning occurs in the context of collaborative work
- the students assume responsibility for the learning.

(summarised by [24])

The restructured ASD II course provides student with a number of opportunities:

- to identify, analyse and solve a number of issues, repetitively. This acts as preparation for professional employment
- to practise the art as well as science of SE in a laboratory setting
- to test the understanding of theory, its connection with application, and develop theoretical insight
- to deal with incompleteness and ambiguity
- to think independently and work co-operatively, fostering insight into individual strengths and weaknesses.

Despite their initial misgivings, students agree (some a little reluctantly) that the course is a worthwhile experience (one week 11 comment *you need more practical application of the theory you teach (ASD II style)*…).

From the teacher’s perspective, exposure to the uncertainties, inconsistencies and idiosyncrasies associated with real problems helps ensures students exits the educational environment with the potential to deal in their own turn with wicked problems, and hence become competent software engineers within an organisational context.
References


