Student perceptions of quality learning: evaluating PBL in Software Engineering

Jocelyn Armarego, School of Engineering Science, Murdoch University

During 2003 a PBL environment was introduced in the Software Engineering degree program in an attempt to address student preconceptions regarding the way they should learn, and enhance student creative potential, a skill considered highly desirable within the discipline.

This paper describes PBL environment provided in the context of the redevelopment of the Requirements Engineering unit, looks at the student cohort undertaking the unit, and describes their reaction to this alternative way of learning. A review of the qualitative feedback indicates that further research, in particular in student approaches to study, and an analysis of student learning style could go some way in explaining the student reaction to the PBL environment.

Introduction

The School of Engineering Science provides a number of degree programs focussing of the development of software. *Requirements Engineering* (RE) is the first of the core Software Engineering (SE) units, currently offered in semester 1 of the second year of study. During their first year students have been immersed in a scientific/engineering paradigm where problem-solving through laboratory procedure, repeatability of experimentation and rigour in mathematics is a key learning objective. As has been described previously (Armarego & Clarke, 2003) RE provides a contrast to the learning environment provided in their first year that some students find difficult to assimilate. Although due process and procedure has its place, the focus of the unit is on divergent thinking and the development and evaluation of alternatives. In this unit they are asked to ignore the problem-**solving** (coding) of a situation presented, and to explore and then formulate the problem itself.

Qualitative feedback and anecdotal evidence indicate that students' expectations of the learning environment are challenged by this change in focus:

- they expect there to exist a definitive solution to the (known) problems with which they are presented (à la science/mathematics)
- they expect to define the problems only in terms of the programming language with which they are familiar (currently Java)
- they expect a fundamentally competitive class environment to exist
- they expect their 'wild ideas' to be laughed at and ultimately rejected, and therefore are inhibited in expressing them.

In summary they see software development as fundamentally scientific, where following a defined process will lead to a quality product (Pfleeger, 1999), only requiring enhanced technical competence. The expectations noted above are in keeping with the engineering stereotype revealed by an analysis of student learning styles.

Since its inception in 1999 the unit has been taught in workshop mode. All content was made available online, so lectures and tutorials were replaced by discussion, exercises and group evaluation of alternatives presented. The prime motivation in changing from this learning environment to PBL (problem-based learning) was to address the issues noted above and to reduce the level of teacher direction that was identified as a concern in a review of the course based on Reeves (1997).

Establishing a creative PBL environment, based on authentic problem scenarios and creativity-enhancing processes seemed an appropriate approach for both challenging student preconceptions and focussing on the insight-driven opportunistic nature of the RE process (Carroll & Swatman, 1999; Guindon, 1989; Visser, 1992). The background to the redevelopment of the unit (including the review mentioned above) has been previously discussed (Armarego & Clarke, 2003). This paper looks at the student cohort undertaking the unit, and describes their reaction to this different way of learning.

Context for learning

Establishing an environment

The PBL environment focuses on the secondment of the class to a (virtual) organisation – collaboration between a software house and the university. *MurSoft* requires a team to work, on short-term placement, on a project to develop gaming software to be used as an educational resource within a tertiary institute. This provides an authentic context for learning: students will have an opportunity, within their final year of study, to undertake an internship with a software-based organisation.

In order to ensure the team will integrate well, the students are initially provided with a very small problem to define. This problem introduces students to the *MurSoft* environment, and also serves the purpose of introducing the PBL process. Students are given some little time to familiarise themselves with other members of the class (since the rest of semester was to be spent on collaborative tasks) and with the lecturer, who takes on the role of academic consultant (not the client, but a resource students have access to). All interaction with the client is undertaken through web-based material: memos, minutes of meetings, telephone messages, 'talking heads', press releases etc provide the problem triggers required. Triggers act as prompts to students to undertake some task identified in the PBL process.

Unit content is centred on a text and online material, which act as a constraint: students initially explore this material in order to achieve the learning outcomes they have identified in a problem component, rather than having unlimited access to resources on the Internet and elsewhere. This is a significant issue: RE is a relatively new discipline, with varying approaches taken in its description. It is important at this early concept-learning stage that students are not confused or frustrated by the presentation of too many alternate viewpoints, tools, definitions for the same concept etc. This is likely to occur if students are to explore freely during the *self-directed learning* stage of the PBL process. Again, providing environment constraints adds to the authentic approach: as graduates, students will be expected to follow the operating procedures standardised within the employing organisation.

Benefits versus cost

Much has been written regarding the value of PBL in learning, eg (Boud, 1985; Wilson & Cole, 1996). However, undertaking such a project comes at a cost:

- *content* – guidelines for implementing PBL indicate that success is partly based on a reduction to the content covered: assuming too much content is a pitfall in a PBL environment (Albanese & Mitchell, 1993)
- *time to develop project* – Bridges (1992) suggests that each PBL project requires 120-160 hours to construct, field-test, and revise. To this figure should be added technical effort when the problem is developed in an online environment
- *cost* - PBL is economical for classes of less than 40 students (Albanese & Mitchell, 1993). It does not scale well to large student numbers without great increase in staffing resources
- *more time to teach less content* - Albanese & Mitchell (1993) suggest 22% more time is required to teach in PBL mode, despite the reduction in content usually advocated. In their study, when academic staff consider the time per week in preparation to teach problems in comparison to presenting lectures, instead of 8.6 hours/week primarily preparing lectures, staff spend 20.6 hours/week primarily in groups with students
- *difficulty in transitioning*, both for staff and students - Bridges (1992) suggests academic staff are uncomfortable withholding information as they watch students struggle with problems, and need training to develop facilitator skills or they may be unsuccessful in PBL. Students may be uncomfortable with the extensive collaboration required or with the lack of teacher-direction given.

A team comprising unit co-ordinator, the academic support staff member from Murdoch's TLC (Teaching & Learning Centre), a third year software engineering student as research assistant and a computer science student with expertise in interactive graphics as technical assistant laboured over one semester plus the summer break to redevelop the unit. Final testing was completed only days before the unit commenced: this tight schedule meant that some decisions were deferred (eg the lead time was too short to remove the final exam from the assessment profile). Absolute commitment to run in PBL mode was not required until after Week 1 classes – workshop mode as backup was always possible. However, once the decision had been made it was very important that the technical infrastructure was robust: the problem scenario was dependent on online triggers being released automatically and available to students.

Major 'intellectual' effort went into deciding on minimum content – the question became one of what did the students have to have to be effective in further units. Although this is an issue in all PBL environments, it had added significance in RE because it is foundational for the SE degree program – many discipline-specific concepts are introduced here. In the end, a focus on creative strategies for dealing with ill-structured problems was as important as content. While acquiring particular domain knowledge remains one of the unit objectives, adaptiveness in generalising knowledge in order to enhance productive thinking as a basis for insight and true novelty of thinking is equally important. Productive thinking is the ability to use past experience on a general level, while still being able to deal with each new problem situation in its own terms. Gott, Hall, Pokorny, Dibble and Glaser (1993) posit that this adaptive/ generative capability suggests the performer not only knows the procedural steps for problem solving but understands when to deploy them and why they work. However, the development of productive thinking requires learning effort spent on abstraction and reflection in addition to content.

How well did we do?

Both formal and informal assessment was undertaken over the semester: data may be categorised as qualitative or quantitative:

- quantitative assessment
    - the major assessment of the unit was based on group work (three components)
    - the exam modelled previous exams, and was based on questions that had been used before, so in theory it was possible to compare how well students performed in comparison to previous cohorts
    - two individual components (a Performance Review and a Portfolio)
- qualitative assessment
    - in-semester year surveys - the year co-ordinator asks for comments/problems regarding all the units undertaken over the semester. These surveys are conducted within the Engineering discipline in week 4 and 11
    - students completed an end of semester unit assessment –this is University-based
    - as noted above, one of the final components of their formal assessment was to prepare for a Performance Review. As well as some more technically based issues (eg how easy would it be to go to design from the specification developed by your team) students were asked for their impressions on their team performance and asked to comment on whether they thought they learnt less or more this way.

Quantitative assessment

The results achieved by these students will not be described here, except to note that, as shown in Figure 1, the PBL environment did not appear to unduly disadvantage the students.

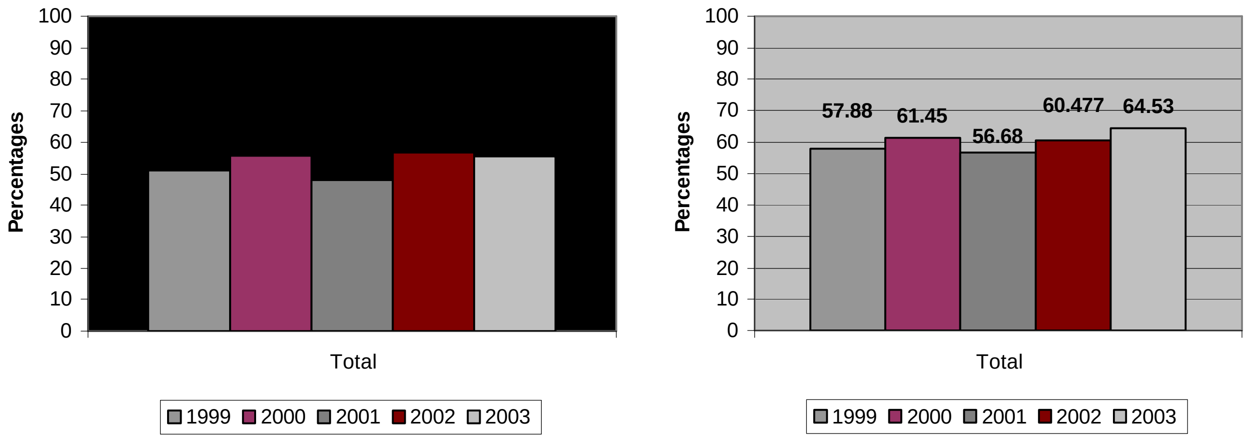


Figure 1 RE 1999 - 2003 (a) Average raw exam mark (b) Average final mark

Qualitative assessment

The Engineering discipline within the School informally surveys all students within each year group to identify general problems that are both unit-specific, and that relate to the mix of units undertaken. Students are asked to identify good and bad points during Weeks 4 and 11 of semester (ie usually near the first point of feedback and towards the end formal classes).

As the list of representative comments below shows, some elements considered 'bad' by the students (eg *learning by doing*) are a highlight of the PBL process. This may be a reflection of student approach to study or preferred learning style, and deserves further investigation.

Week 4

Good:
–"helps with thinking about all areas of a problem(good for other units)"
–"interesting, practical, well presented"
–"it's really good"
Bad:
–"very vague on assessment and what specifically needs to be completed"
–"inability to work alone"
–"no lecture or tutorial"
–"don't really like how it's structured"
–"don't know what is going on"

Week 11

Good:
–"learn what you like at your own pace"
–"more practical training & real time example"
–"probably useful"
–"easy to get help for unit"
Bad:
–"objectives sometimes unclear"
–"learning by doing"
–"only get the general idea and concept of unit later in semester"
–"not very structured"
–"hard to determine what we are supposed to be working towards"

As a part of the "Performance Review" component of the coursework students were asked to comment on the unit by addressing the following questions:

- what is good/bad about a unit structured this way. Chart 1 shows the results. While some students could appreciate the authentic nature of the environment, this learning style did not sit well with all
- things to add/change/delete in how the virtual secondment and project were organised. Chart 2 shows that students felt they lacked guidance in the form of examples and exercises to use for benchmarking their performance. In addition, they expressed concern that such a large component of their assessment was based on group work
- reflecting on whether they (individually) felt they had learnt more/less this way. As can be seen from Chart 3, the class was fairly evenly divided on this point.
  Students who felt they had learnt less commented on a *lack of mastery of subjects*: (*less every time new content arrives*); of *focusing on components addressed by the project*, on *delegating* and *relying on others* for concepts. Towards the end of semester, some of these students still felt *lost and confused*: *self teaching is not one of my fortés* stated one student, perhaps with a hint of despair.
  Students felt they learnt more in the areas of research, communications (confidence to speak up; need to be heard & get ideas across) and team skills. They added concepts easier to grasp; forced to learn more for project relevant components; and, finally they had to grapple with various perspectives from others. In summary there were: ample resources & up to us to take it.

Chart 1 Recoded comments from Performance Review: good/bad things about a unit structured this way

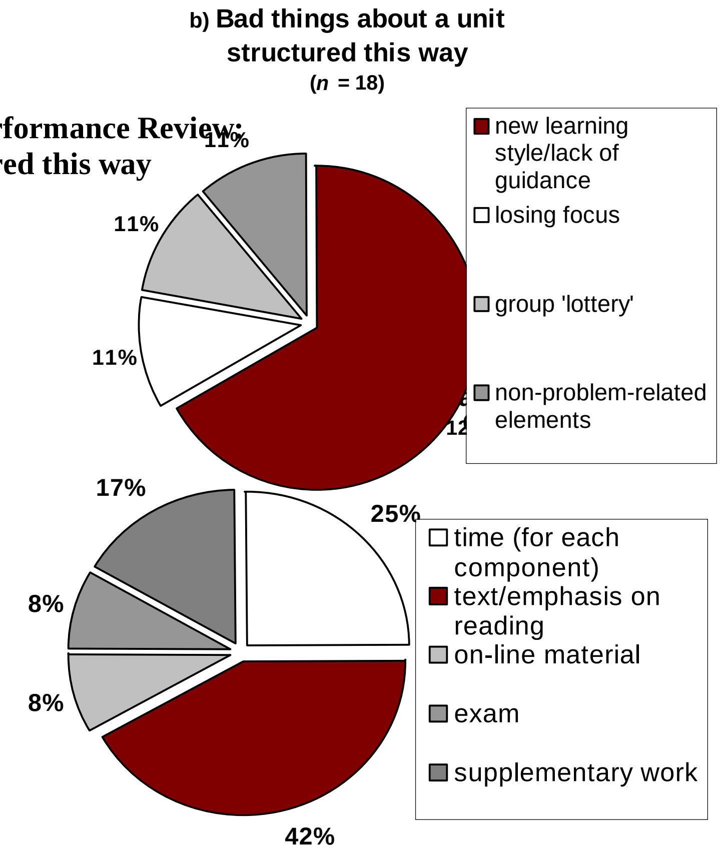
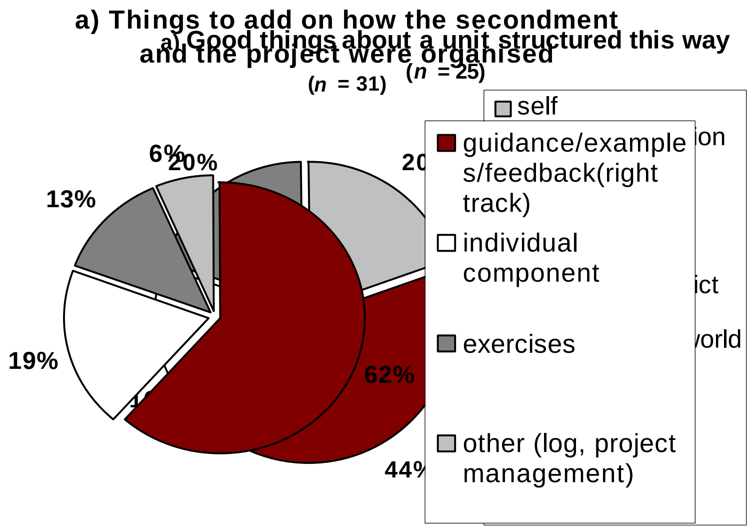
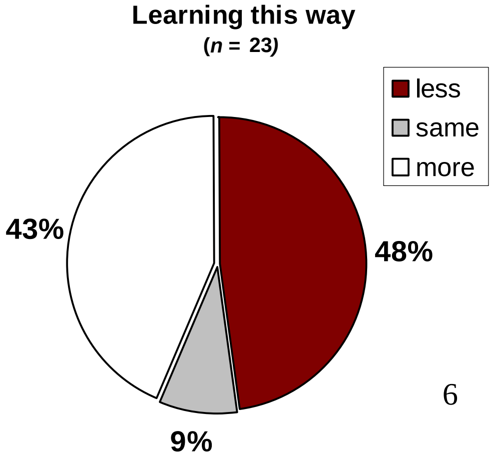


Chart 2 Recoded comments from Performance Review: items to add/change/delete

Chart 3 Recoded comments from Performance Review: learning more/less this way

This last comment characterising the opposing perspectives is perhaps the most pertinent, and an aspect that requires further research. An understanding of the learning motivation for the students might reveal a relationship between why they learn (for meaning or for reproduction)





and how they expect to learn.

This feedback also shows that, although a great deal of effort went into preparing the PBL environment, more scaffolding is required. Students need greater preparation in order to tackle a different learning model (eg a better understanding of the PBL process), and support structures (examples, etc) so that they have a clear indication of the appropriateness of their learning.

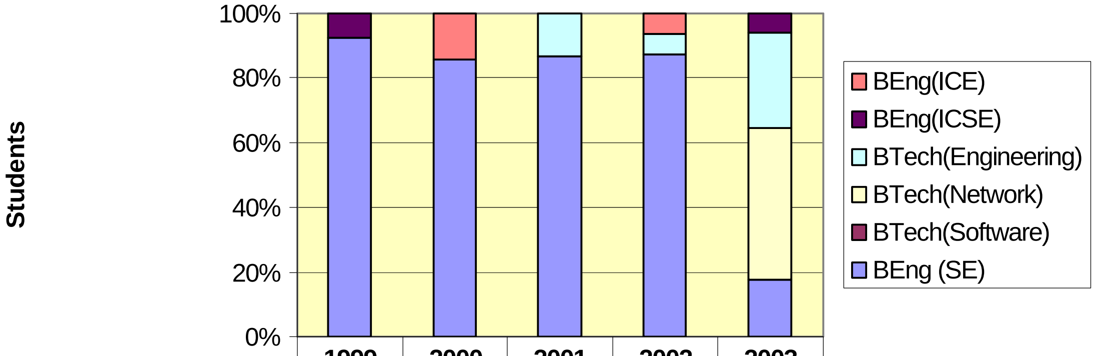
Characteristics of the class

In previous years, the class predominantly comprised students expecting to complete a 4-year B Engineering degree, most probably in software. These students come into the program with a higher tertiary entrance score than B Tech students, and have completed one year of study within the School. This has implications in learning expectations, acceptance of the learning culture within the School, etc. As Figure 2 shows, in 2003 the B Eng students were in the minority. An additional factor in 2003 was that almost 50% of the B Tech students were TAFE (Technical and Further Education, ie technical college) articulation students. These students are entering the program with advanced standing, and although a formal analysis has not been undertaken at this stage, there is anecdotal evidence that they had additional problems, which could be summarised as an expectation to be taught. A discrepancy between the Teaching/Learning environment they were expecting and the one they were experiencing was almost immediately apparent. Ferris (2003) provides a discussion of the issues of advanced standing in an engineering education environment.

Other research undertaken within the School looks at learning styles (Fowler, Allen, Armarego, & Mackenzie, 2000). Table 1 compares RE students (unit code ENG260) with students show strong bias in each of the dimensions: they are *active*, therefore should prefer group work and dislike lectures, *sensing*, therefore prefer to learn facts, and solve problems by

Figure 2 *Requirements Engineering* class cohort 1999 – 2003

a well-established process, strongly *visual* where reading text is classed as verbal and fundamentally *sequential*, implying a preference for logical steps to problem-solving (Soloman & Felder, 1999).



	1999	2000	2001	2002	2003
BEng(ICE)	**0**	**1**	**0**	**1**	**0**
BEng(ICSE)	**1**	**0**	**0**	**0**	**1**
BTech(Engineering)	**0**	**0**	**2**	**1**	**5**
BTech(Network)	**0**	**0**	**0**	**0**	**8**
BTech(Software)	**0**	**0**	**0**	**0**	**0**
BEng (SE)	**12**	**6**	**13**	**14**	**3**

Table 1 Soloman & Felder *Index of Learning Styles:* 1999-2003 cumulative results

Clients		**Input**	**Understanding**
	REF	**VIS**	
Eng 1st yr	44%	77%	
ENG108 2003	52%	79%	
ENG 260 2003	**25%**	**95%**	
A/C 1st yr	35%	76%	
CS/IT 1st yr	51%	84%	
Eng 4th yr	24%	86%	
Eng Staff	73%	73%	

The Kolb (1984) results support these findings: RE students are classically engineering types. They are pragmatists (*convergers*) who revel in active experimentation (labs, fieldwork) with a tendency to narrow technical interests, or theorists (*assimilators*) with a forté in the basic sciences. The imaginative ability of the *diverger* or the intuitive problem solving of the *accommodator* is sadly limited. Yet these are skills highly rated in a Requirements Engineering context.

Table 2 Kolb Learning Style Inventory 1999 – 2003 cumulative results (percentages)

	Eng 1st year	ENG 108 2003	ENG 260 2003	A/C 1st year	CS/IT 1st year	Eng 4th year	Eng Staff
Accommodator	8	4	10	13	5	3	0
Diverger	18	8	10	13	12	7	17
Assimilator	33	42	40	47	56	38	41.5
Converger	41	46	40	27	27	52	41.5

Where to from here

The PBL environment these students have experienced may be considered a creative one: one of the aims of its development has been to enhance divergent thinking and the creative potential of students. It would seem, however, that such an environment does not match the learning characteristics of the student cohort. Tracking the cohort through subsequent units will go some way to confirming (or not) the value of PBL in engineering. This is critical in the context of a strategic move away from traditional lecture/tutorial/lab-style learning within the discipline area at Murdoch. Future research into student approaches to study will provide some insight that will assist in further offerings of this unit and of others within the engineering degree programs.

References

Albanese, M., & Mitchell, S. (1993). Problem-based learning: a review of the literature on its outcomes and implementation issues. *Academic Medicine, 68*(1), 52-81.

Armarego, J., & Clarke, S. (2003). *Preparing students for the future: learning creative software development - setting the stage.* Paper presented at the Learning for an Unknown Future: Proceedings of the Annual International HERDSA Conference, Christchurch (NZ).

Boud, D. (1985). Problem-based learning in perspective. In D. Boud (Ed.), *Problem-based Learning in Education for the Professions* (pp. 13-18). Sydney: Higher Education Research Society of Australasia.

Bridges, E. M. (1992). *Problem based learning for administrators* (No. ERIC Document Reproduction Service No. ED 347 617). Eugene (OR): ERIC Clearinghouse on Educational Management.

Carroll, J. M., & Swatman, P. A. (1999). *Opportunism in the Requirements Engineering process* (School of Management Information Systems Working Paper No. 1999/02): Deakin University, Australia.

Ferris, T. L. J. (2003). *A technician is half an engineer - overseas students and advanced standing in engineering degrees.* Paper presented at the Engineering Education for a sustainable future: proceedings of the14th Annual AAEE Conference, Melbourne.

Fowler, L., Allen, M., Armarego, J., & Mackenzie, J. (2000). *Learning styles and CASE tools in Software Engineering.* Paper presented at the Flexible Futures in Tertiary Teaching. Proceedings of the 9th Annual Teaching Learning Forum,, Perth.

Gott, S. P., Hall, E. P., Pokorny, R. A., Dibble, E., & Glaser, R. (1993). A naturalistic study of transfer: adaptive expertise in technical domains. In D. K. Detterman & R. J. Sternberg (Eds.), *Transfer on Trial: intelligence, cognition and instruction* (pp. 258-288). Norwood (NJ): Ablex.

Guindon, R. (1989). The process of knowledge discovery in system design. In G. Salvendy & M. J. Smith (Eds.), *Designing and Using Human-Computer Interfaces and Knowledge Based Systems* (pp. 727-734). Amsterdam: Elsevier.

Kolb, D. A. (1984). *Experiential Learning Experience as the Source of Learning and Development*, : Prentice-Hall.

Pfleeger, S. L. (1999). Albert Einstein and empirical software engineering. *IEEE Computer, 32*(10), 32-37.

Reeves, T. C. (1997). *A Model of the effective dimensions of interactive learning on the World Wide Web*. Paper presented at the Proceedings of Interaktiivinen Teknologia Koulutuksessa (ITK '97), Hameenlinna (Finland).

Soloman, B., & Felder, R. (1999). *Index of Learning Styles (ILS)*,, from http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html

Visser, W. (1992). Designers' activities examined at three levels: organisation strategies and problem-solving processes. *Knowledge-Based Systems, 5*(1), 92-104.

Wilson, B. G., & Cole, P. (1996). Cognitive teaching models. In D. H. Jonassen (Ed.), *Handbook of Research for Educational Communications and Technology* (pp. 601-621). New York: Simon & Schuster Macmillan.

Acknowledgement

Sally Clarke, formally of the Teaching & Learning Centre, Murdoch University and currently at University of Auckland for always asking questions.