48433 Software Architecture

Reflections on Spring 2004

John Reekie
Faculty of Engineering
University of Technology, Sydney

December 2004

Contents

1 Preliminaries

This short report contains the reflections of the instructors and students concerning the delivery of the subject 48433 Software Architecture in the Spring 2004 semester.

The 2004 design document is [1]. (A new design document for 2005 will be forthcoming.) Previous reflections for this subject were published for Autumn 2004 [2].

2 Personal reflections on Spring 2004: JohnR

This was an interesting semester, with some excellent outcomes and some that were not so great. More details below...

Lectures

The lectures were just "OK." Perhaps it was something to do with the terrible room that we were put in, but I felt that the class overall was not very responsive. The class did liven up towards the end of the semester, and in the presentations given by students in the last two sessions I was very pleased to see them take the initiative to find different topics to present to the rest of the class.

The lecture materials were improved over last semester, although not hugely. I didn't cover as much ground as I really wanted to. Nonetheless, considering the relative immaturity of software architecture instruction at the undergraduate level, I felt that we (me and the class) still did pretty well.

One thing I feel very positive about is that the subject is extremely well positioned for *next* semester! I think I have figured out now exactly how to present software architecture in a way that makes sense to inexperienced developers, and so next semester the pace will be faster and we'll get to some more serious architecture towards the end of the semester.

Course materials

I spent a great deal of time writing course materials, in the form of chapters of my so-called book. This has addressed the major complaint from Autumn 2004: lack of adequate lecture notes. For 2005, I plan to have the book draft printed prior to the start of the semester.

Labs

This semester there were four, two-hour labs, to introduce the students to some simple architectural concepts in a hands-on way. The labs covered: simple concepts of client-server computing and threading; an introduction to programming for databases; a simple three-tier web system; and an introduction to realtime audio. There was supposed to be a middleware lab using CORBA, but I decided that the realtime lab was more important this time around. A number of students used RMI in their final project anyway, but unfortunately with little appreciation of its (considerable) liabilities.

I'm very pleased with the outcomes from the four labs, considering the relatively low investment. I can see that the quality of the programming assignment,

and the students' understanding of the concepts of incorporating off-the-shelf software into their designs and systems, have vastly improved compared to previous semesters.

Blogs

In Autumn 04, two students wrote blogs, which I really enjoyed reading. As a result, in Spring 04 I offered all students the opportunity to write a blog on softwarepractice.org. (It was an option for the "participation and communication" mark. The other option was to give a 15-minute presentation to the class.)

Regrettably, this experiment did not work as well as I had hoped. My intention was that the blogs would act as an "electronic learning journal." Those that used it that way used it very well, and I once again enjoyed reading their blog entries. Other blogs contained little reflection and a lot of pointless complaining. For next semester blogs will again be a bonus mark, and will have to be approved by me based on a written proposal containing the first entry.

I also wonder whether the blogs tended to work against activity on the discussion board. The board is hard enough to get going anyway, but this semester you could have found more life in Dawn of the Dead! I suspect that perhaps students who might otherwise have posted on the board wrote in their blog, and then the "ownership" aspect of a blog tended to inhibit followup discussion form other students.

Assignments

This semester I moved some of the programming portion of the assessment into a "throwaway prototype" about 2/3 of the way through the semester. This was not very successful. I'm not sure why, but I think perhaps I will just ditch it for next semester.

The final assignment, the programming assignment, I haven't marked yet (as of this writing). From what I have seen so far, it looks like next semester I will need to provide a working build structure and (perhaps) insist on everything being checked into CVS or subversion.

The written assignments went fairly well, although I think I emphasized conceptual architecture too much again. For next semester, I will compress conceptual architecture into the first assignment. Execution architecture is where the students seem to have the most difficulties anyway.

I am also thinking about a different assessment scheme. Instead of three assignments, have three milestones and a final deliverable. The problem with this idea is that it moves marks to further back in the semester.

I'm slightly disappointed that none of the teams bothered to find out any background about the software systems used in their assignments. I thought the system narratives I provided them were quite interesting and thought that students would be motivated to explore a little further than they did. I wonder if I can in future provide some incentive for a little more originality and exploration.

Exam

I didn't have a mid-term exam this semester, but I think I will go back to one next semester. That will help the students with a little motivation to work through the exploration packs and readings a little earlier, and will get ten marks out of the way early (-ish) in the semester.

At the point of writing, I have marked about two-thirds of the final exams. I have to say that I am very pleased overall with the work most students submitted for the exam. Next semester I think I will expand the "design" section of the exam, and restructure the format to give a good bit more time for that section.

Concluding remarks

As I mentioned, I feel that the subject is now positioned very well for future semesters. There is still a fair bit of work to do in preparing the "final" version of the course materials, but once that's done it should run very smoothly from now on.

I am looking forward to teaching it with someone else next semester, though. This semester I did it solo, which is a bit of a drag.

3 Numeric survey results

3.1 Subject feedback survey (SFS)

Here are the numeric results from the University-run subject feedback survey (SFS) conducted towards the end of the Spring 2004 semester. The number of responses to the survey was 24.

The scale used for answers to these questions is:

1. Strongly disagree
2. Disagree
3. Neutral
4. Agree
5. Strongly agree

- Table 1 summarizes the overall results for Spring 2004 and includes the questions.
- Figure 1 shows the results compared to the Faculty and the University. Every question is now at or above the Faculty average, except for questions 3 and 5, which clearly need some attention next semester. Several are well above the University average as well.
- Figure 2 shows the history of SFS results over the last three semesters.
- Figure 3 shows the distribution of results for Spring 2004.

Nr	Question	Mean	SD
1.	This subject was relevant to me	4.4	0.5
2.	The subject was delivered in a way which was consistent with its stated objectives	4.0	0.4
3.	I had a clear idea of what was expected of me in this subject	3.0	1.1
4.	My learning experiences in this subject were interesting and thought provoking	4.1	0.8
5.	I found the assessment fair and reasonable	3.3	1.0
6.	There were appropriate resources available to support the subject	3.6	1.0
7.	I received constructive feedback when needed	3.6	1.0
8.	Overall I am satisfied with the quality of this subject	3.7	0.8

Table 1: Subject Feedback Survey results, Autumn 2004

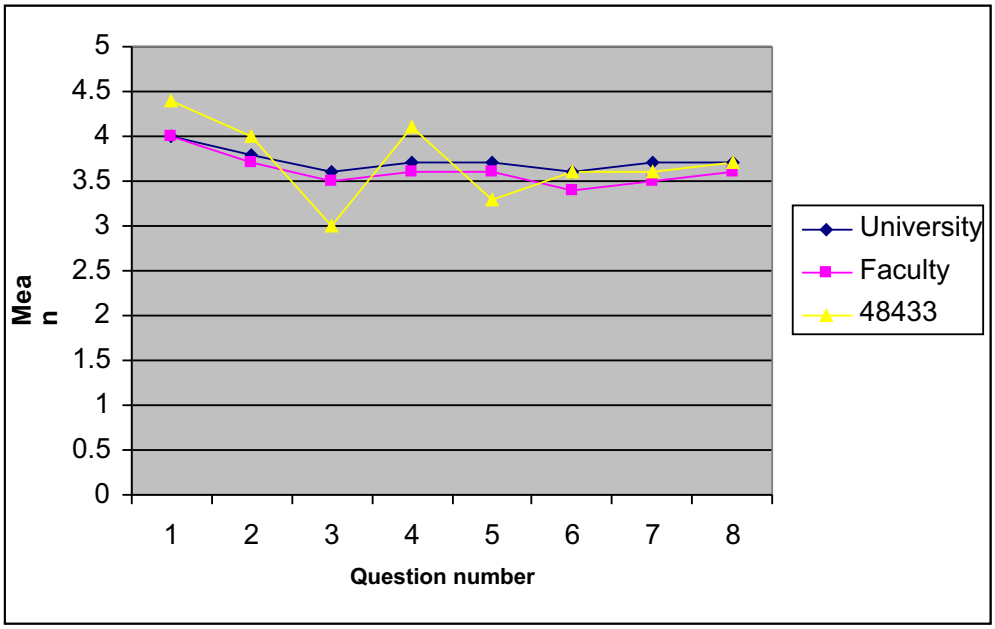

Figure 1: SFS means compared to the Faculty and the University

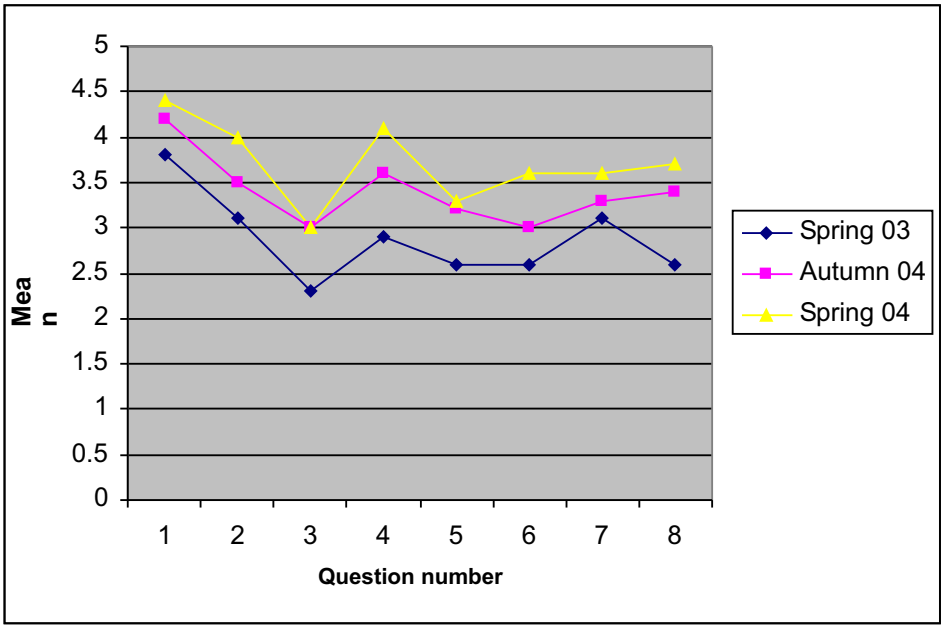

Figure 2: SFS means over the last three semesters

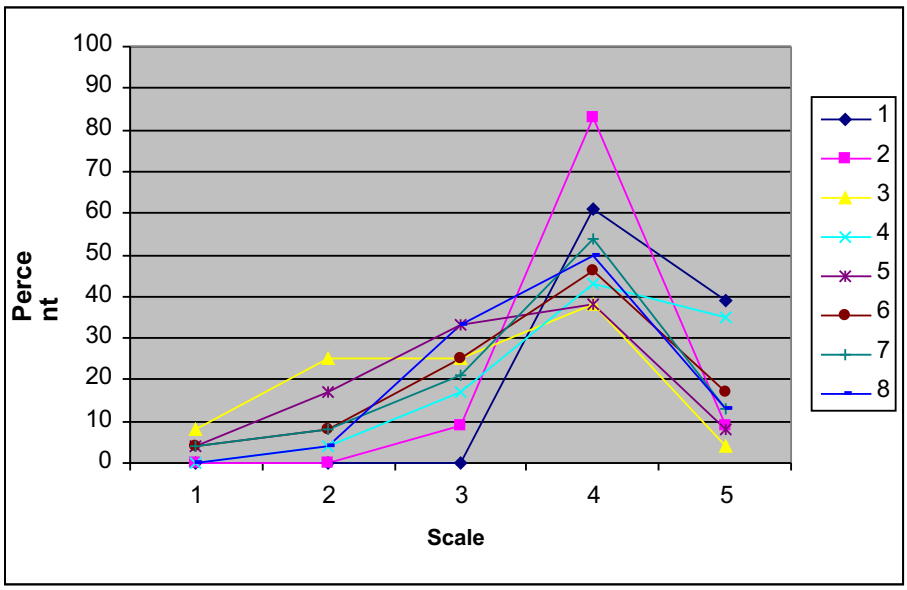

Figure 3: Distribution of SFS responses

Nr	Question	Mean	SD
1.	The subject met its designed quality of *Relevance*	4.3	0.6
2.	The subject met its designed quality of *Fairness*	4.1	0.7
3.	The subject met its designed quality of *Enjoyability*	4.3	0.6
4.	The inspections improved the quality of my assignments	3.9	0.7
5.	I will use inspections again in other subjects or projects	4.0	0.6
6.	I expect to use what I have learnt in this subject in other subjects	4.2	1.0
7.	I expect to use what I have learnt in this subject at work	3.7	1.0
8.	The degree of difficulty of this subject was appropriate for this stage of the course	4.0	0.6
9.	There was a good balance between theoretical and practical concerns	4.4	0.7
10.	The workload was appropriate for a subject at this level	3.6	1.0
11.	The group assignments improved my ability to work as part of a team	3.2	1.0
12.	Guest lecturers made a valuable contribution to the subject	3.6	0.9
13.	I improved my ability to think critically	3.6	1.0

Table 2: Informal Feedback Survey results, Spring 2004

3.2 Informal feedback survey (IFS)

I also gave students an informal survey that I collected and collated myself. The purpose of this survey was to collect more specific data than provided by the University SFS. The survey was requested after the teaching semester was over, and included responses up until submission of final results. There were 11 respondents.

- Table 2 lists the questions asked, and the mean of the responses for each. Again, answers ranged from 1 (Strongly Disagree) to 5 (Strongly Agree).
- Figure 4 is the means plotted for this semester and the last.
- Figure 5 shows the responses given by students for the numbers of hours spent on the subject for each third of the semester. It is not clear from the responses whether this time includes the time spent in class, as the question was not clear about this.

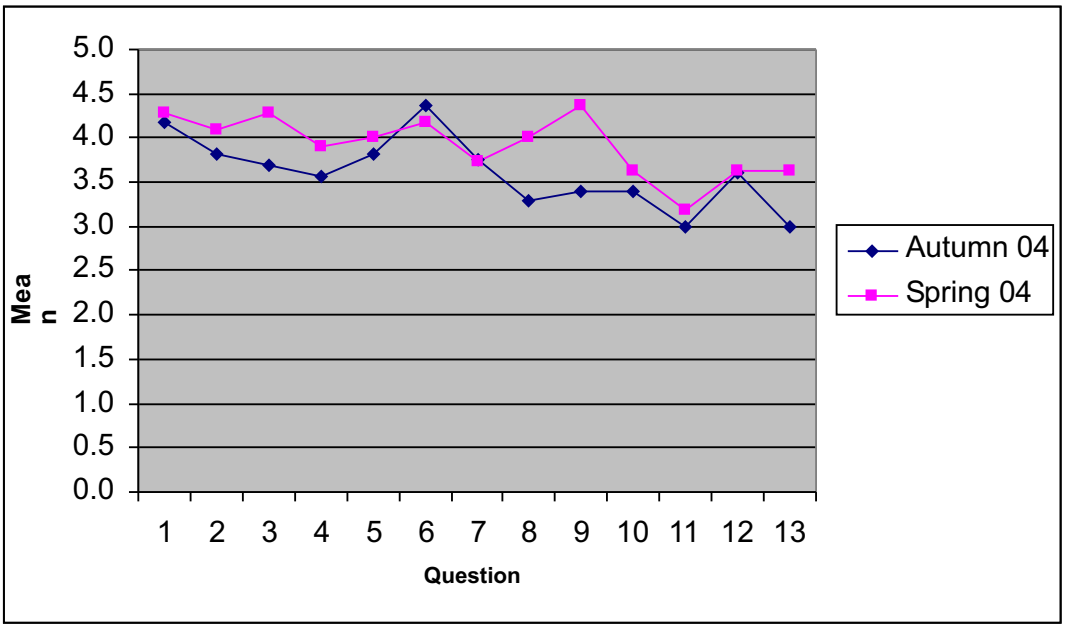

Figure 4: IFS response means

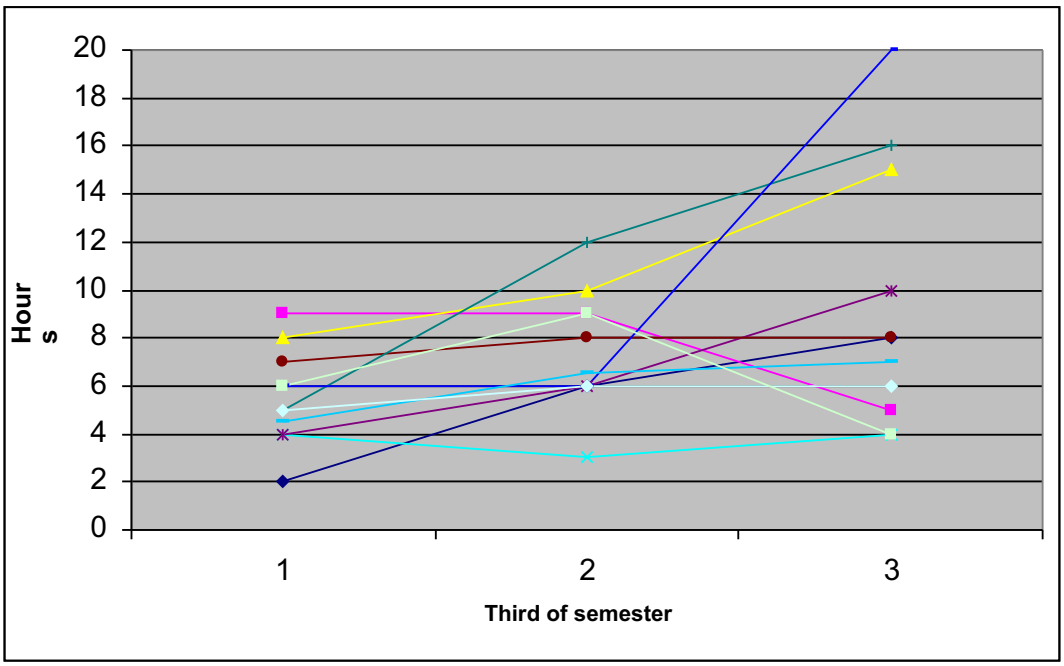

Figure 5: Weekly workload

3.3 Discussion

I'm very pleased (and relieved!) that the student perception of the subject has improved significantly this semester. In the University-run survey (Table 1), questions 1, 2, and 4 are now well above the University and Faculty. Questions 6, 7, and 8 have improved a lot and are now on a par with the University. Questions 3 and 5 have stubbornly remained where they were last semester, well below both the University and Faculty.

While nobody should design a subject just to improve SFS results, it's a useful indicator and I obviously need to pay some attention to the reasons for the lack of improvement in these two questions. In other words, why were students unclear of what was expected of them in the subject, and why did they feel that the assessment was not fair and reasonable? I'm unclear about what the real issue is here.

In the informal survey, all questions remained the same or improved, except for question 6, which decreased slightly. I'm not sure if the difference is significant, but it is slightly worrying that "I expect to use what I have learnt in this subject in other subjects" has gone down. I will see how this goes next semester.

I'm not quite sure what to make of the students' response to the workload question. (I had the same difficulty last semester.) The trend to increasing workload as the semester progressed remains, despite my efforts to flatten it out. Perhaps this is just human nature! In the first two-thirds of the semester at least, the workload seems reasonable for a 6 credit-point subject. In the last third, about a quarter of the students blew out their time allocation for the subject. The means are, however, only slightly higher than last semester at 5.5, 7.4, and 9.4 (Autumn 04 was 5.6, 6.2, and 9.1).

4 Written student feedback

4.1 Informal surveys

The informal survey asked the following:

> *Please include a short reflection on the subject. Any items of particular note that you would like to comment on would be well-received. Was there anything about the subject that you thought was particularly good? Anything particularly bad? Was there anything that you learnt that you thought particularly interesting? Do you have specific suggestions for improving the subject?*

Responses received (in no particular order):

- This subject was taught in a structured manner with each of the architectural analysis and design requiring the completion of an assignment. This helped in solidifying the concepts from the lectures. The laboratory tasks also helped with the construction of the prototype. This subject seemed to be quite intense compared to other subjects from experience. 3 assignments, 1 laboratory assignment, and 1 prototype plus the final exam seemed to be a lot of work in my opinion. The only suggestion that I would make would be to remove the laboratory assignment. This to me

seemed a little redundant as the final prototype covered the topics from the labs. Overall the subject was well taught and certainly provided me with useful tools which now allow me to approach the architecture and development of software in a different manner with a different mindset.

— Patrick Poniros

- Overall happy with the subject. You were refreshingly informal and grounded. One comment is that perhaps it may have been a good idea to give an overview of the process of software architecture, before jumping into it stage by stage. I personally perceive things better when they are presented top-down.

  Also because implementation architecture was not covered it wasn't particularly clear how the architecture would convert into a detailed design. It seemed that (other than the conceptual architecture which was fairly detailed) we were just scratching the surface most of the time. That really left the system a bit abstract.

  Quote on this very same topic (approximate): "Keep things as simple as possible, but not any more" (Einstein). I think that last bit might have been slightly violated in the subject.

  Finally speaking for myself and a couple of my fellow classmates the major problem is that it was difficult to allocate any significant time to the subject due mainly to SSD. Hence the proposal to make this subject a prerequisite for SSA is a worthy one.

  Thanks John - overall it was a 4-star subject. It's kind of like the movie you liked and might have bought it on DVD, but not enough to bring it up in conversations every chance you got.

  Later

  — Max

- One thing that made me screw up my assignments was that I didn't read the handed readings/texts carefully. The texts are easy to understand for students at any level and therefore should be taken seriously. After finishing some of my assignments, I went through the texts again and discovered that I really overlooked some important issues, which I could have caught if I had studied the texts. I would therefore recommend future students to study the text wholeheartedly. Regarding the labs, I reckon they should be made harder so that students will feel the threat towards them, thus making them study harder. Maybe the database sections and client/server can be combined into something bigger and push towards the beginning of the semester, as the lecturer said. After that web-server maybe a good thing. Anyway, the thing is the lab should be at 3rd year level, although it doesn't mean they should be too hard or focused too much on technology. Finally, I think it maybe easier for student if they know what are expected in the final prototype, because the prototype takes a bit of time to finish and redundant features will eat up precious time, therefore affecting the main architecturally significant features.

  — (Name withheld)

- The subject was relevant to me as a SE student. The workload is quite in balance between theory learning and practical works. Lectures' contents were interesting but more guest lecture was expected. All the labs were useful except lab 4. The student should be made clearer of to what extent they were required to develop the throwaway prototype.

  — (Name withheld)

- Reflections reflections reflections! I don't like how I had to write so many relections throughout the subject!

  This subject has been thought-provoking and well worth the work. I have learnt a lot throughout the semester, and John Reekie is not so bad - despite rumours indicating the opposite ;)

  The only criticism that I can offer is that there are too many assignments worth too few marks. It's a lot of work, for little reward. I suggest reducing the assignment workload in some way. Perhaps make it only one prototype, and make it due earlier in the semester, rather than so close to the exams.

  Overall, definitely one of the better subjects in the SE major.

  — Daniel

- This subject reflects a vast improvement in the UTS Software Engineering degree and in my opinion, is a step in the right direction. There are a few subjects still in this degree that most Software Engineers would find completely irrelevant. If these can be weeded out and replaced with more relevant subjects such as Software Architecture, the entire degree could possible double in its value to the students as well as industry.

  Unfortunately, due to circumstances I had to do it at the same time as SSD while it would have been far more valuable to me to have completed it before SSA itself. I feel that Software Architecture should most definitely be a pre-requisite for SSA/SSD.

  The subject clarified and cemented my approach to software architecture and also shifted my paradigm on some aspects. The practical side of the subject is also very welcome and I am quite pleased that we did in fact realise the concept of architecture in some real, tangible software code.

  I do however feel that the subject could include more on real-world architecture solutions and patterns. In particular, we could cover what is middleware and the different types and forms of middleware available. Another suggestion would be to place a little less emphasis on web-based systems and perhaps consider some other domains where software is pertinent. Some areas for consideration include real-time needs, and portable and embedded devices. We could also do a case study and discuss the architecture of an industry-grade technology/product - to see how its done in the real-world, and also to critically analyse real-world things because they were, after all, built by some other human beings with human as well as circumstantial constraints.

  Hope this helps to improve what is already an excellent subject.

  — Dinesh Gurrram

4.2 Formal surveys

The University-run Subject Feedback Survey has two open-ended questions at the end. I suggested to the students filling in the survey that they wait until the informal survey, but many chose to enter comments anyway. In future it would simply be better to just let the students know that they can use either method to provide additional feedback.

1. *What did you particularly like in this subject?*

    - The material and views presented are cutting edge, yet based on reality – well grounded.
    - It's been emotional.
    - Good support, good resources.
    - It was related to my industry. It gave me more of an understanding about my chosen profession.
    - The subject matter. Good solid academic.
    - The tutorials help a lot.
    - Lectures and interaction with the lecturer.
    - Lecturer's style and knowledge.
    - The importance of Software Archutecture and processes and techniques to be used for it.
    - The team-based assignment, emphasis. As the real way to learn SA is to undertake it in a team-based environment.
    - Doing the prototype.
    - Not sure.
    - Relevance to work/career objectives
    - New concepts!

2. *Please suggest any improvements that could be made to this subject.*

    - N/A
    - Less B.S. [Unreadable next sentence]
    - Less assignments.
    - Clear requirements for the assignment. More examples in the lecture about important issues.
    - More labs and greater participation in them.
    - Less assignment.
    - Better feedback and clear answers to questions on the web. Some of the material available at lecture was not available at any other time.
    - Coding assignments were not useful. Methodology used and assignments were vaguely defined. Needs a few more iterations.
    - Lower workload. Less emphasis on programming.

- Resources — I'm guessing in future years a textbook will be available which will be extremely useful.
- The assessments were subjective, was hard to know the requirements in order to get a good mark.
- Marking criteria for assignments, some assignments required significant amounts of time for a few marks.
- Too many assignments.
- Better lecture room. Better assignment guidelines.
- Clearer idea of expectations.
- Fewer assignment! Lab assignment is not necessary, the project (prototype) is enough.

5 Concluding remarks

While some things didn't work as well as I had hoped, overall the results and feedback from the subject are very positive. The subject has made progress the student cohort overall was reasonably satisfied with the subject. I'm confident that there is further improvement coming next semester—it's "merely" a matter of completing additional lecture material, book chapters, and assignment guidelines, as well as sorting out some of the burning issues evident in the surveys and student responses to surveys!

My thanks to all that contributed to this report.

References

[1] John Reekie. The architecture and design of 48433 software architecture—autumn 2004 version. Online at http://www.eng.uts.edu.au/ johnr/pdf/sa-design.pdf, October 2003.

[2] John Reekie and Lian Loke. 48433 Software Architecture: Reflections on Autumn 04. Online at http://www.eng.uts.edu.au/ johnr/pdf/autumn04.pdf, July 2004.