

# 48433 Software Architecture

Reflections on Autumn 2004

John Reekie, Lian Loke  
Faculty of Engineering  
University of Technology, Sydney

July 2004

# Contents

<b>1 Preliminaries</b>	<b>3</b>
<b>2 Reflections on Autumn 2004: JohnR</b>	<b>4</b>
<b>3 Reflections on Autumn 2004: Lian Loke</b>	<b>6</b>
<b>4 Numeric survey results</b>	<b>7</b>
4.1 Subject feedback survey (SFS) . . . . .	7
4.2 Informal feedback survey (IFS) . . . . .	7
4.3 Conclusions drawn . . . . .	11
<b>5 Written feedback</b>	<b>14</b>
5.1 Feedback received . . . . .	14
5.2 Conclusions drawn . . . . .	14
<b>6 Changes for Spring 04</b>	<b>16</b>
6.1 Delivery modules . . . . .	16
6.2 Assignments . . . . .	16
6.3 Labs . . . . .	16
6.4 New Subject Guide . . . . .	16
<b>A Written student feedback</b>	<b>21</b>
A.1 Student blogs . . . . .	21
A.2 Discussion board . . . . .	21
A.3 Informal surveys . . . . .	21
A.4 Comments in formal surveys . . . . .	27

# 1 Preliminaries

This document contains the reflections of the instructors and students concerning the delivery of the subject 48433 Software Architecture in the Autumn 2004 semester.

This document also contains incremental improvements to the subject that will be performed for the Spring 2004 semester.

This document is an incremental update to the first version of the design document [1]. The next iteration of this document (for Autumn 2005) will be a complete rewrite of the design and architecture document.

## 2 Reflections on Autumn 2004: JohnR

I quite enjoyed teaching this subject in Autumn 2004. I think overall the students learnt more and enjoyed the class more than in the Spring 2003 semester (when the subject format was completely different). I'll be the first to admit that I kinda ran out of gas a little over halfway through the semester. Creating a new subject from scratch with no existing subjects as a model and no suitable textbooks is quite a task. I am either doing something brilliant here, or I have completely gone off the deep end :-)

### Lectures

Overall, I'm pretty happy with the way the lectures turned out. I had huge amounts of time leakage on the preparation and was unable to complete lecture materials for all modules, but I think it's a solid start. The positive student feedback (later sections) indicates that the work put into preparing the lecture materials was worth it.

To prepare the lectures, I created a "story" for each, consisting of twenty or so points, each of which makes up one slide. I've found this to be a very useful technique for planning and revising lectures. There are stories for most modules written, with some stories for revised lectures done as well.

### Attendance

I didn't require attendance in classes. I tried to make it clear that attendance in class is expected and that marks will be negatively impacted by not attending. The only formal "penalties" for not attending were if not present at an inspection (maximum of half marks), missing the exam (obviously, no marks), and not attending the final presentation (fail the subject).

In spite of the general lack of coercion for attendance, I felt attendance at classes was very good. I did not do any headcounts to prove it, though.

### Course materials

Again, even the fairly sparse "lecture notes" for the subject petered out a little over halfway. I'm being much more sensible next semester, and have reduced the number of modules from twelve to ten. I'm also trying to get several chapters of "my book" into shape for use in the class next semester.

### Programming exercises

I was disappointed that very few students did the optional exercises, in particular those that required compiling and running a simple program. Those exercises were intended to illustrate important concepts. So I've decided to further abuse my already-abused budget for this subject and develop a set of labs and a lab assignment for next semester. (So much for being sensible.)

### The Big Gap

I suppose it should be obvious, but I was actually surprised that some students didn't start to properly study the material and do the exploration packs until

after the delivery of the subject was over! This is much too late to start asking questions, and I think my curt responses on the board put some students offside.

For next semester, I will try and make my expectations clearer in the Subject Guide and in the classes. When I've finished delivering the subject, it's over for me. Maybe I will pretend that I've gone on holiday already.

### **Overall**

Well. In the end, I have to say it was quite a blast to deliver a brand-new subject “on the edge,” so to speak. Having Lian co-teaching and co-developing the subject material was a tremendous help. I'd really like to work this way in future semesters if at all possible.

I'm also very grateful to all the students that provide written feedback on the subject, and in particular the two students who wrote blogs throughout the semester—this provided feedback that you just don't get while up in front of the class.

### 3 Reflections on Autumn 2004: Lian Loke

In my role as the offsider to John, we worked quite effectively on development of the course material. Being somewhat of a novice myself on the subject, I could assume the students' position and this helped to clarify and refine the presentation of material, particularly the lectures and exploration packs. Working together in this way provided an inbuilt review mechanism and allowed for sharing of ideas and learning.

John's approach to teaching the subject was well grounded in practice. The set of 4 assignments allowed the students to experience an iterative, evolutionary development of a design. From marking Assignment 3 (the online education system), some students grasped the concepts of qualities and architectural behaviour well and were able to reason about their choices. Others struggled with identifying relevant components: some were distracted by the web interface and placed most of the functionality as a "blob"; some had far too many components; some chose weird names and created components called the "backend" - whatever that does. The use case maps were quite well done, but then the connection to the concurrency view was often poor or not well explained. The concept of concurrency definitely needs more work - perhaps the lecture and lab material can be revised for concurrency.

John and I had talked of presenting the lectures together, but given the time limitations (particularly of myself) this didn't really happen. In the future (once the lecture material is more stable) this could add even more pizzazz!

In terms of subject development, the structured system that John uses for building the material is commendable. I particularly liked the "story" approach to developing lecture slides. His system makes this process visible - better than scribbling things down on disjoint pieces of paper that get forgotten.

All in all, it was excellent teaching with Dr Reekie — thank you!!

## 4 Numeric survey results

### 4.1 Subject feedback survey (SFS)

Here are the numeric results from the University-run subject feedback survey (SFS) conducted towards the end of the Autumn 2004 semester. The number of responses to the survey was 28.

There are eight questions in the standard University survey. I also added another six questions (numbered 9 to 14—next semester I will probably just do those in my own informal survey, see below). 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 and includes the questions.
- Figure 1 shows the results for the standard eight question. Shown are last semester (Spring 2003), this semester (Autumn 2004), and the Faculty of Engineering and the University as a whole for Autumn 2004. This semester's results are significantly improved, but are not quite up to the Faculty average yet, with questions 3 and 6 being the farthest below.
- Figure 2 shows the distribution of results for the previous semester, Spring 2003.
- Figure 3 shows the distribution of the results for Autumn 2004. Compared to the previous semester, the number of "Strongly Disagree" responses has decreased, and the number of "Strongly Agree" responses has increased.
- Figure 4 shows the means obtained for the additional six questions. Figure 5 shows the distribution of responses.

### 4.2 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. There were 15 respondents.

- Table 2 lists the rankable questions asked, and the mean obtained. Not all of these questions are intended for the purposes of obtaining a ranking.
- Figure 6 is the means plotted (this will be more interesting next semester...)
- Figure 7 shows the distribution of responses to the questions.

Nr	Question	Mean	SD
1.	This subject was relevant to me	4.2	0.6
2.	The subject was delivered in a way which was consistent with its stated objectives	3.5	0.9
3.	I had a clear idea of what was expected of me in this subject	3.0	1.2
4.	My learning experiences in this subject were interesting and thought provoking	3.6	1.0
5.	I found the assessment fair and reasonable	3.2	1.1
6.	There were appropriate resources available to support the subject	3.0	1.1
7.	I received constructive feedback when needed	3.3	1.1
8.	Overall I am satisfied with the quality of this subject	3.4	1.0
9.	The degree of difficulty of this subject was appropriate for this stage of the course	3.3	0.9
10.	There was a good balance between theoretical and practical concerns	3.4	0.9
11.	The workload was appropriate for a subject at this level	3.4	1.0
12.	The group assignments improved my ability to work as part of a team	3.0	1.2
13.	Guest lecturers made a valuable contribution to the subject	3.6	1.0
14.	I improved my ability to think critically	3.0	1.1

Table 1: Subject Feedback Survey results, Autumn 2004

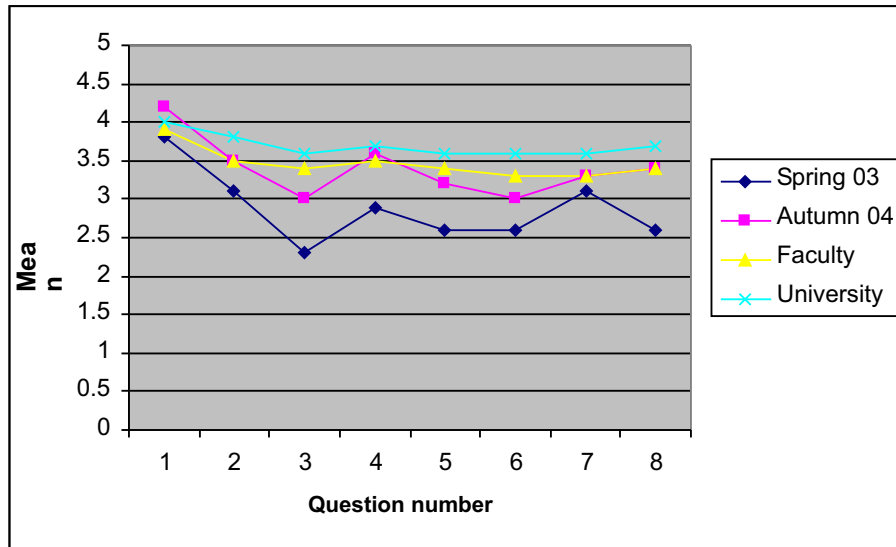


Figure 1: SFS means for 48433 compared to the Faculty of Engineering and the University as a whole, Spring 2003 and Autumn 2004

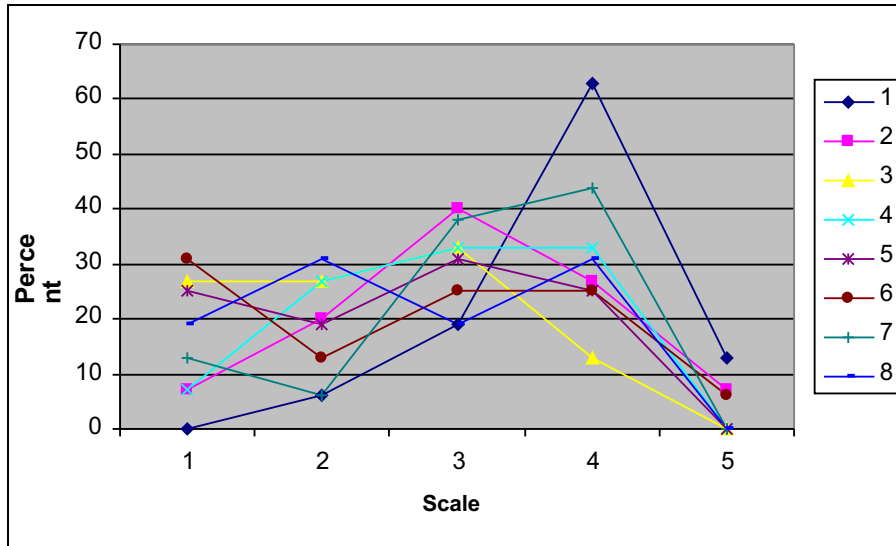


Figure 2: Distribution of SFS responses for the previous semester, Spring 2003

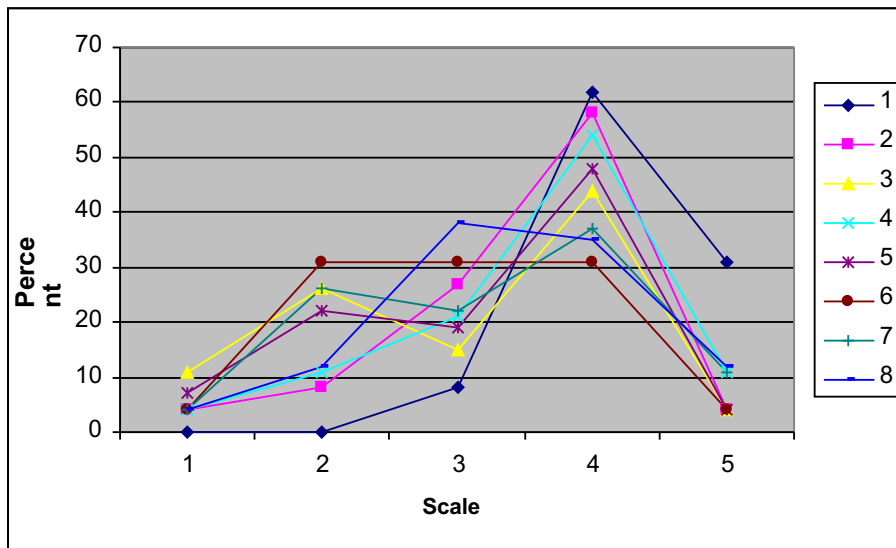


Figure 3: Distribution of SFS responses, Autumn 2004

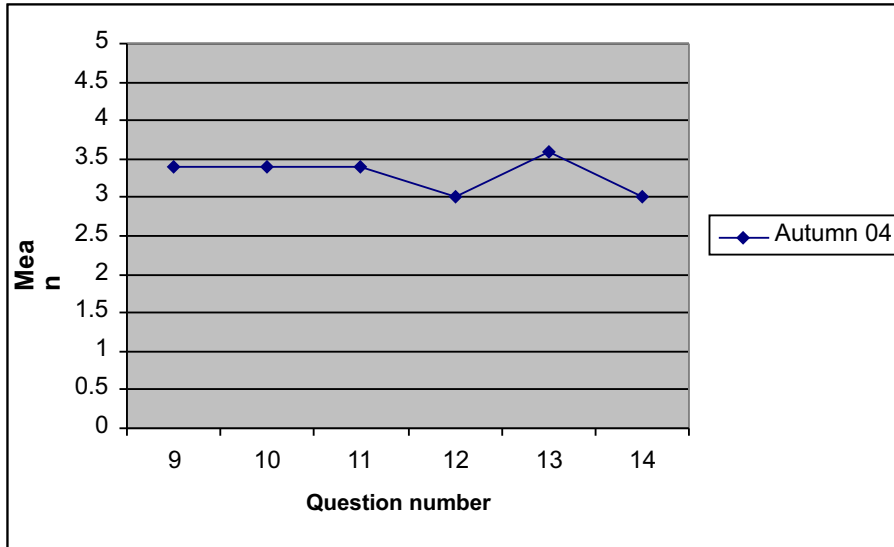


Figure 4: SFS means for 48433, additional questions

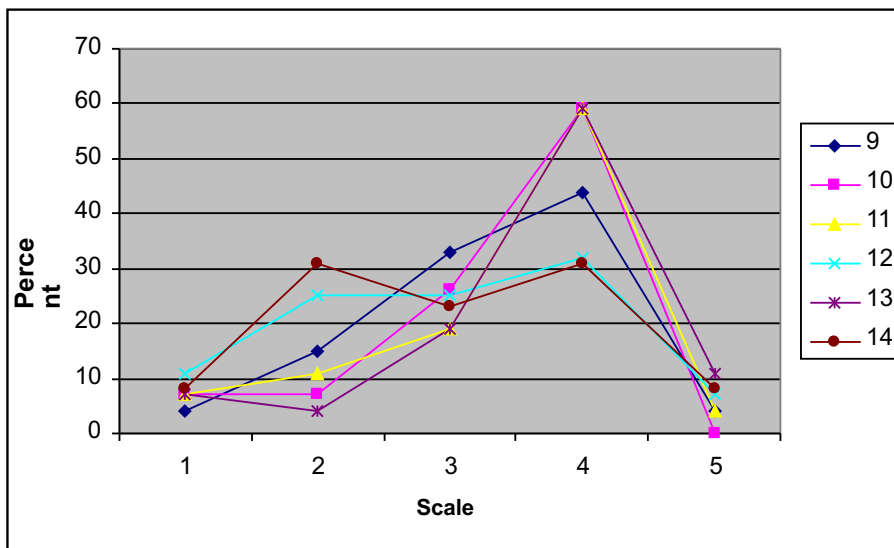


Figure 5: Distribution of SFS responses, additional questions, Autumn 2004

Nr	Question	Mean	SD
1.	The subject met its designed quality of <i>Relevance</i>	4.2	1.0
2.	The subject met its designed quality of <i>Fairness</i>	3.8	0.8
3.	The subject met its designed quality of <i>Enjoyability</i>	3.7	0.7
4.	The inspections improved the quality of my assignments	3.6	1.1
5.	I will use inspections again in other subjects or projects	3.8	1.1
6.	I expect to use what I have learnt in this subject in other subjects	4.4	0.7
7.	I expect to use what I have learnt in this subject at work	3.8	1.2

Table 2: Informal Feedback Survey results, Autumn 2004

- Figure 8 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. At any rate, two things are apparent:
  1. There was a large spread in time spent on the subject, and
  2. Students either spent a constant time over the semester, or tended to increase their time towards the end of semester.

### 4.3 Conclusions drawn

I draw the following conclusions from the numerical survey data.

- Overall performance
 

Overall, the subject has moved from well below the Faculty average to just below. This is very encouraging. I did of course hope for a slightly better result this semester, but all things considered I'm very pleased with the overall result.
- Workload
 

I'm slightly confused about whether the subject as run in Autumn 2004 has a low, a fair, or an unreasonable workload.

One thing that was apparent from the figures is that the workload tended to increase as the semester progressed. For Spring 2004, I have therefore moved some assessment further forward in the semester to try and spread the workload more evenly.
- Expectations and resources

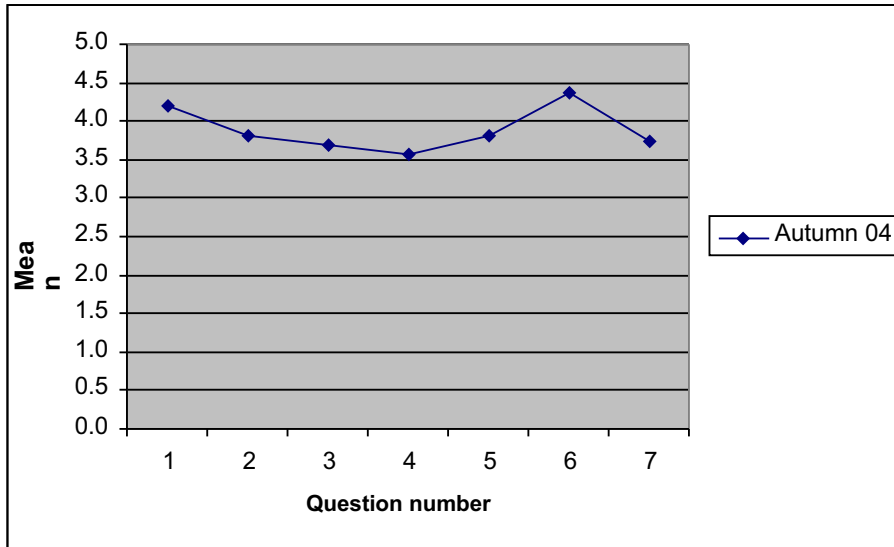


Figure 6: IFS response means, Autumn 2004

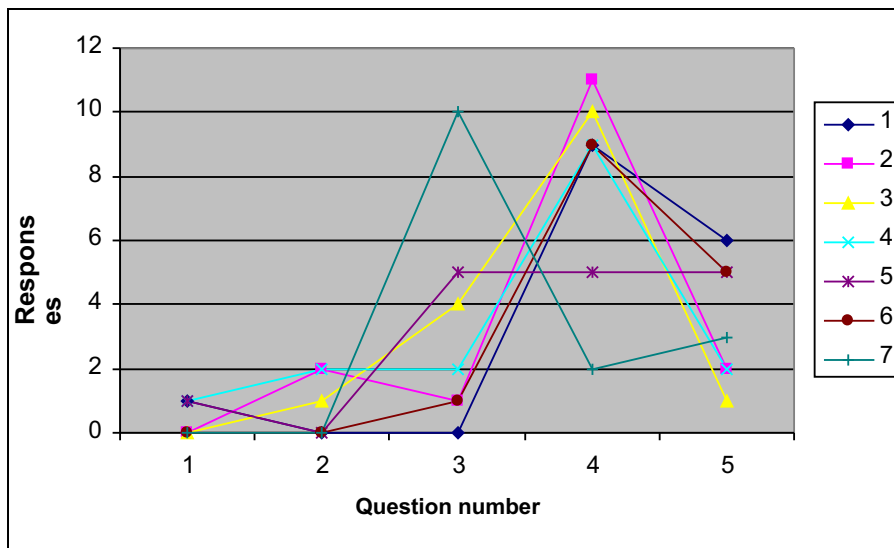


Figure 7: Distribution of IFS responses, Autumn 2004

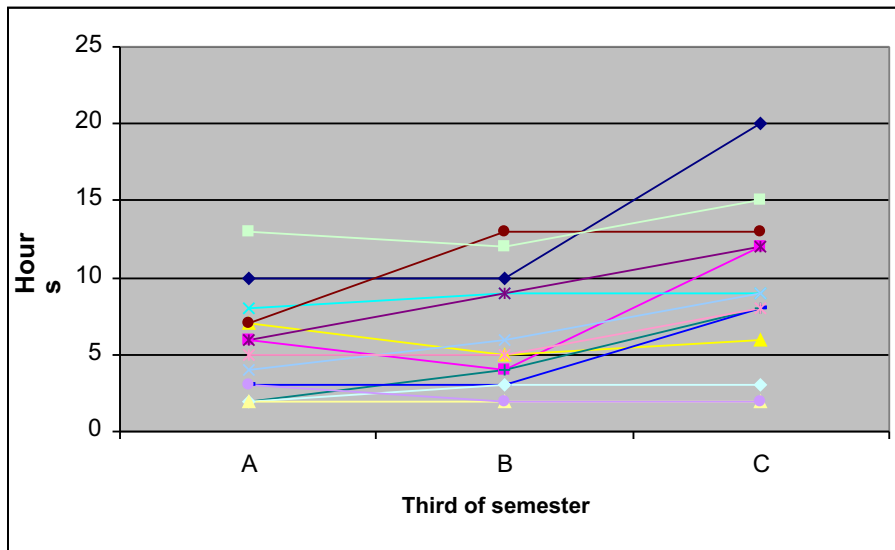


Figure 8: Weekly workload, Autumn 2004

These two items were the low-performing elements in the SFS, and so need to be addressed specifically for next semester. They correspond to the questions:

- I had a clear idea of what was expected of me in this subject
- There were appropriate resources available to support the subject

The latter item is being addressed, at least in part, by the book I am trying to write. I am not sure yet how to address the former.

- Relevance

Although the question whether the subject met its Relevance quality attribute received a favorable mark, the response to the question on whether students expected to use knowledge from this subject at work was decidedly underwhelming. I don't know why this is, but this needs to be watched for next semester.

- Guest lecturer

A favorable result! Good job, Rohan!

## 5 Written feedback

### 5.1 Feedback received

Written feedback was received from students by the following means:

- Student blogs
- Discussion board
- Comments in the formal University-run feedback survey
- Reflections in the informal feedback survey

*All* written feedback for Autumn 2004 is reproduced (or linked to) in Appendix A.

I very much appreciate the effort by students to provide constructive feedback on the subject, especially those who wrote blogs and reflections in the informal survey. Despite the numbers obtained from the surveys in the previous section, I feel that the thoughtful written reflections provided by these students is perhaps the most helpful. Thanks folks :-)

### 5.2 Conclusions drawn

From reading Appendix A, I draw the following main conclusions:

1. Lectures and presentations were well-received.  
All students who mentioned lectures seemed to enjoy the presentation material and style.
2. The “formal” inspections met with mixed opinions.  
In general, they seemed to be well-received in principle, but not so well in the way they were implemented in this subject. Specific suggestions made by students, which I agree with entirely, are to:
  - have inspection teams see the material beforehand
  - have inspection teams be working on the same software system

For Spring 2004, I have reduced the number of inspections from four to two, and will attempt to implement the above two suggestions. In the longer term, I need to confer with my colleagues in the SE program to decide where the best place to do inspections properly would be. I suspect that Software Engineering should introduce the concept, Software Architecture should do a couple simple ones, and Software Systems Design should do “proper” ones.

3. Supporting materials were not as good as they could have been.  
One of the main difficulties I have had with this subject is the complete lack of suitable texts to support it. I suppose it’s possible that I’m going about this all wrong, but I don’t think so.... I think that it’s just not a mature field and that there are no good undergraduate textbooks.

Therefore, I am writing the first one :-)

One student commented that the supplementary readings were not related to the subject material. This simply highlights the difficulty with teaching this subject at this level. The readings were related (I am fairly certain) but you have to get well past the basic material presented in the subject to get a lot out of them.

#### 4. Examples

This item is a specific instance of the previous one. Students wanted more (or any, actually) worked examples. The problem is... there aren't any! Pick up any book on Software Architecture and you will see what I mean. Plus, it takes time to develop good examples, especially in Software Architecture. As the subject proceeds through the next few iterations, the number of examples will build up.

#### 5. Software Architecture should precede Software Systems Analysis.

As presented in this subject, it certainly should! We are working to get the pre-requisite chain altered so that the flow of subjects from OOP, though OOD, SE, SA, and then SSA and SSD, is a lot more explicit in this major.

#### 6. Assignment structure

A number of students commented favorably on the structure of the assignments, and the iterative work that they encouraged. I'm hoping these favorable comments continue next semester, since I've reorganized the assignments to try and improve understanding of the architectural design process a little earlier.

#### 7. Feedback on assignments

Some students complained that they did not get enough feedback on their assignments. I don't understand where this complaint is coming from, as there were plenty of sessions I scheduled in the LDC where only one team showed up to discuss their assignments. And not one team posted any of their assignment afterwards on the board to discuss it.

Students who scheduled an LDC time to see me or (in some cases) approach me in class got plenty of feedback on their assignments. I will try and make it clearer in the Subject Guide that more feedback on assignments will be provided by LDC and discussion board.

#### 8. Discussion board

There seems to be some dissatisfaction with a perceived over-reliance on the discussion board. I am not sure what to do about this one. I see the board as an effective way for students to get additional interaction with the instructor that they can't get any other way. I will attempt next semester to better explain both the time (aka budget) constraints we are under, and present the board as a "free" adjunct to the regular (but limited) means of consultation.

## 6 Changes for Spring 04

This section describes changes to be made for the Spring 2004 semester. Since there is not much time between the Autumn and Spring semesters, changes have to be incremental. The changes chosen to be implemented are based on my own observations during the semester, and on the student surveys and feedback summarized in the previous section.

### 6.1 Delivery modules

The number of delivery modules has been reduced from twelve to ten. This allows the necessary time for revision, for students to complete projects, and for exam preparation, towards the end of semester. It also provides some “buffer” in case of unexpected events like strikes and power outages.

An unavoidable consequence was that the content of a number of modules was also rearranged. I’m hoping that the better flow of the early part of the subject justifies the extra work re-organizing slides and other materials.

Figure 9 shows the new module structure overload in the subject concept map. (The concept map has been elaborated slightly since Autumn 04.)

### 6.2 Assignments

The number and type of assignments has been changed. In particular, the final written assignment has been removed and a new “lab” assignment has been introduced in its place. The Lab assignment and the first written assignment can now be done individually only.

The delivery of assignments has been pushed further forward in the semester, both to reduce end-of-semester marking load, and to even out students’ workload during the semester.

Figure 10 summarizes the coverage area of the written assignments.

### 6.3 Labs

Four labs sessions are being introduced in Spring 04. This is to “encourage” students to do the hands-on exercises that are so important to their understanding of a reasonable subset of the material. Three other sessions are scheduled, for assessment of the two programming assignments, and for help with the final project assignment.

The coverage of the lab sessions is illustrated in Figure 11.

### 6.4 New Subject Guide

I have written a new Subject Guide, basically from scratch. The main purpose is to clarify student expectations of the subject, in areas such as:

- Access to instructors
- Use of the online discussion board
- Structure and difficulty of the final exam
- Expected class attendance



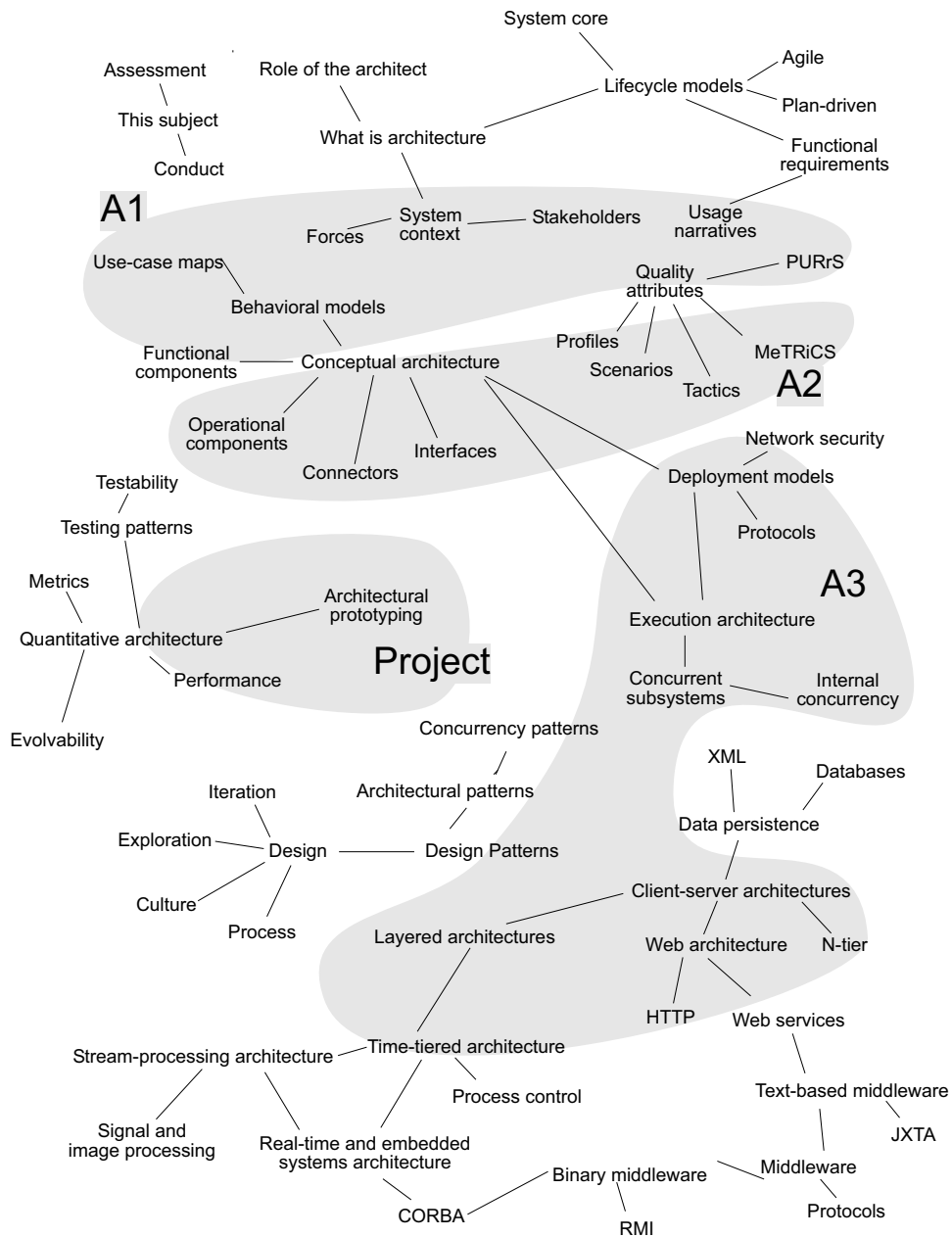


Figure 10: Topic areas of this subject, showing assignments



A secondary goal was to convert the Subject Guide from Word to Latex. Word is essentially unmaintainable, and I decided that maintainability was a more important goal than compatibility.

## A Written student feedback

This appendix provides all written student feedback for the Autumn 2004 delivery of the subject.

### A.1 Student blogs

Two students wrote blogs (“weblogs”) over the course of the semester. They can be viewed at:

- [http://www.bastardblade.com/mt/ren/archives/cat\\_software\\_architecture.html](http://www.bastardblade.com/mt/ren/archives/cat_software_architecture.html) (Renaldy Bodhiwan)
- <http://www.theartofsoftware.org/?q=blog/15> (Anh Phan)

### A.2 Discussion board

There were a number of threads on the discussion board that provided feedback during the semester. In particular, one thread after the exam indicated that many students were surprised by the expected difficulty of the exam, and were concerned about the “must pass” requirement for the exam. (After marking the exams, I decided to waive the “must pass” requirement, both for this semester and in the future.)

The discussion board can be viewed at:

- <http://www.softwarepractice.org/viewforum.php?f=3>

The thread expressing concern over the exam can be read at:

- <http://www.softwarepractice.org/viewtopic.php?t=237>

### A.3 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?*

Responses received:

- Software Architecture is pretty new and good area that software engineer should be aware of. The subject has provided a lot of new useful information such as business requirements, non-functional qualities and architectural strategies to achieve them.

At conceptual level, some concepts are quite hard to understand. It might be good if you give students “full” examples of all the qualities, examples about profiles, scenarios. They might stick to a system, or different systems that you use through the semester. VI is good, but it doesn’t cover enough to help students understand at deeper level.

What about architecture other than "conceptual"? and other aspects such as architecture analysis, software reconstruction, software migration?. Should they be covered in this subject? They haven't been covered in other subjects that I've learned so far.

— *Xuan Thah Dang*

- I liked the way the course was designed. I could see that a lot of effort had gone into the preparation of notes, exploration packs and slides. However, I noticed that they tended to become sparser towards the end of the semester.

The design of the assignments was good. I liked how we were encouraged to evolve our architecture over numerous iterations.

The supplementary readings were good (the ones I read), however, they didn't quite fit into the subject.

I enjoyed the opportunity to blog about the subject and get marks on it. This the first time I've kept a blog for a subject and looking back over it, I feel it's a good thing.

The discussion board was a good way to communicate issues outside of the class.

I think I want to be a software architect.

— *Renaldy Bodhiwan*

- The way this subject was conducted was original and interesting. For once we have a subject which seems to not rely on dead text books, but brings in current practice, and is quite flexible. The assessment being based around consistent work throughout the semester to deliver milestones was great. Promotion of open learning as "equals" was also another positive, and the subject was overall a positive experience.

I would have liked to see a little about architecture evaluation methods, and the final 5 mark question in the exam had me stumped. Admittedly I could have prepared for it, but perhaps one more optional question?

— *Lee*

- I thought the subject was entirely relevant, enjoyable and interesting. Every deliverable evolved my perspective on software architecture until the final assignment, changed my entire focus of software systems development. For systems I'm not familiar with (e.g. sockets, n-tier) prototyping is the only way to validate a design. This subject taught me how to identify and categorise desired system qualities and check the prototype satisfies all of them.

This subject taught me the basics of software architecture: the views, qualities, strategies, patterns. The way in which this was taught was mostly satisfactory from my point of view. Monday morning lectures were always informative and interesting, but lacked any concrete information (towards the end of semester) other than some hand-outs and blank/non-existent slides. The hand-outs were useful, but I needed more peripheral information, like with RDBMSs, web-architectures, performance evaluation. The other thing that bothered me was the lack of any real debate (online or

otherwise). John could perhaps understand the lack of confidence engineers (particularly at this uni) have when it comes to communication and maybe start his own threads - with Lian.

For its second run, this was pretty good. I'm sure John will make incremental improvements to his teaching methods, lectures and assignments that make it a better subject for future students.

— *(Name withheld)*

- Good:
  - good detailed and interesting lectures
  - inspections were very useful
  - online forum good aid for discussion on relevant topics
  - very useful subject for SSA and SSD

Improvements:

- would be better if lecture notes were more detailed
- prefer no final exam. Make the assignments more detailed with respective weightings

Interesting topics:

- design patterns
  - embedded systems
- *(Name withheld)*

- Overall, I thought this was an excellent subject. The subject matter was very relevant, and the assignments matched the subject matter very well. My main disappointment was the fact that the subject did not cover any of the formal methods that can be applied to software architectures. In particular, I thought some coverage of Architectural Description Languages (ADLs) would have helped the subject in two ways. Firstly, it is possible that ADLs will find increasing use in industry. Secondly, using ADLs would teach how to approach a software system's architecture in a logical and rigorous manner, which is something I still feel I haven't quite gotten from this subject.

I thought the assignments were very good. I particularly liked the fact that we were told exactly what was expected in each deliverable, while still having some latitude to do what we felt was best. I also liked the fact that each assignment aligned with the subject matter.

In short, this was one of the most worthwhile subjects I have done in the degree.

— *(Name withheld)*

- I thought that the assignments were well-designed and thought provoking. They required a sufficient amount of work to gain an understanding of the subject material without being unnecessarily arduous (like writing 100 page documents!).

The analysis techniques taught were interesting and provided a good foundation for architectural design. It was a shame that I hadn't done this subject BEFORE SSA and SSD, as I feel that it would have resulted in better architectural designs for those subjects. Actually, I think that many students would benefit from doing this subject between software engineering and SSA, as it seems to fit in well there with how I developed my knowledge as a software engineer.

It was fun and the lectures were interesting. Perhaps because I am interested in the subject matter, but I think some of the credit at least goes to JR's lecturing techniques... :)

— (*Name withheld*)

- I found that this subject was different from your typical purely technical subject of which there are many in engineering. Often there were no concrete answers and I found this challenging. It was a good experience to have to look at software from a higher level.

Having to develop a prototype in assignment 4 was tough given the time that we had and the limited knowledge that we have in programming such applications.

One suggestion would be to have more formal tutorial sessions with less students, say no more than 25 students to a tutorial class. However, I understand that this may not be possible due to limited resources for the subject.

If there was no discussion of a particular weekly question/problem or exercise on the website, there was no way of checking our answers to these questions/problems or exercises other than to start a new thread ourself or to ask directly in your LDC time. I felt that there should be more efficient ways of just checking our answers, perhaps you should provide answers or answer guides or at least some form of answers wherever it is possible given the question (it may not be possible for some questions as there is no clear cut or right or wrong answer).

— (*Name withheld*)

- Just two things. Firstly I didn't find the inspections very useful compared to the code reviews we used extensively during my internship. The primary problems I thought were:
  - lack of a moderator,
  - reviewers had no reading time to examine the review object properly,
  - reviewers often had no knowledge of the system they were examining.

I still think reviews are great but they just didn't seem to work well in this context.

Secondly, I felt like we needed to learn more specifics. We spent a lot of time on the process such as how to write profiles. I guess this is important but I would have liked to have learnt all the OO design patterns covered in Gamma's book and the architectural design patterns in the books some of those other readings were taken from. Then I'd feel like I least had a few

tricks or strategies up my sleeve rather than a vague idea of the process. We need specific knowledge as well. e.g. more strategies for each of the qualities. It's a bit like Human Computer Interaction where we learnt how to write user profiles but barely touched user interface guidelines. It's no good being able to understand the problem if you don't have the tools to fix it.

I think a lot of students have a similar attitude. The UTS approach seems to be to teach the principles and let the students work out the specifics because the specifics change very rapidly. That's fair enough but we get frustrated with only ever talking about vague notions. Fifth year software engineers should know what a singleton pattern is by now but we don't.

A quick note about the assignments: the first assignment should not be free range topic and there should not be the option of doing it individually. We expected our topic would be accepted and it wasn't so we had to re-write the first assignment which wasn't fun and was unfair because many other groups didn't have to. Those students who chose to do the assignment individually would have had similar difficulties come assignment 2.

Otherwise I enjoyed subject. The clip art in the lectures was fantastic :). It was good to actually use the UML standards other than class diagrams (although a little more instruction could have helped). The informal approach was very good too (e.g. encouraging hand-drawn submissions, short assignments and simple prototypes) and stressed the fact that architecture is iterative and not to lock yourself in too quickly. Thanks John!

— *(Name withheld)*

- Good: The application of Software Engineering methods in deriving an appropriate final conceptual architecture. For example, the use of use case maps.

Bad: The inspection process may have been unfair. One group may have a good inspection team and another may have a really bad inspection team, and the feedback may not be evenly distributed across the whole subject. Maybe good to have more than one inspection team, or the lecturer to give an official separate inspection time during LDC times.

— *(Name withheld)*

- I don't believe there was enough feedback from the assignments. How are we supposed to improve and progress onto the next stage of the architectural design if the current version is completely flawed (and not knowing what the flaw is)? May be if assessors could just write a short description of what they believe is wrong with the assignments, students will be able to develop a better understanding of software architecture.

Also I believe some notes would be helpful... I think you're already working on this...

— *(Name withheld)*

- I might just embellish on some of my earlier answers instead here if that's OK;

1. The subject met its designed quality of Relevance.  
I think perhaps some of the subject content was well below what I (as a 4th year student) am capable of, though the project was sufficiently challenging for my level in this course. I might not have felt that way if I'd chosen any of the other projects, as I think the domain knowledge required for the RAT project really added some difficulty that I certainly wouldn't have had developing a web system for example.
2. The subject met its designed quality of Fairness.  
Yep, quite happy in that regard, though you never did get back to me about making up the marks I lost for missing the 2nd inspection. ;-)  
My fault too, no matter now.
3. The subject met its designed quality of Enjoyability.  
Yeah, I enjoyed it for the most part, though I really wasn't into the project by the end of the subject due mainly to group work issues and the difficulties we hit.
4. The inspections improved the quality of my assignments.  
No, not in the least. Half the people doing them didn't bother (for our final inspection, we produced our own list of issues because the group we paired with consisted of 1 person asleep, 1 person with NFL, and 1 person not there) Only 1 group we 'inspected' with offered any value, and that was limited, due to their lack of knowledge regarding our project.
5. I will use inspections again in other subjects or projects.  
I would, but not with 'random' people. It adds nothing and it just frustrating for me.
6. I expect to use what I have learnt in this subject in other subjects.  
Already used some architecture development methods from this subject to guide another group project's architecture design. Worked well.
7. I expect to use what I have learnt in this subject at work.  
Possibly, possibly not. Can't see it in my immediate future, but in a different capacity, I probably would pick up some of the techniques.

Finally, my assessment of the biggest difficulty you'd have running this subject is the mix of knowledge of the group. I'm amazed at how little some of the people in that class knew about their chosen industry, whereas others were far advanced. Tough to say where that subject is most appropriately placed. For me, I had little choice about when to do it, as it was introduced in my 3rd year, but it might actually be a better subject if targeted at higher level students? Skip the stuff about 'what is a relational database' and the like. If people don't know the answer to that question by their 4th year, they're in trouble anyway.

But for me, the content in this subject would not have been much use to me until final stage subjects like SSD or this post grad elective I'm doing anyway, so aiming higher might be an option?

— *Ryan Lohan*

## A.4 Comments in 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?*
  - Architecture practices
  - Assignment scheme, interesting lectures
  - Group projects that are incremental
  - The slide presentations
  - Clip-art, readings, patterns
  - Practical assignments
  - It was really good
  - The relevance
  - Relevant assignments
  - This subject was structured quite well in terms of assignments, but needs more time for some assignments, or the load is excessive.
  - Nothing in particular
  - Team work
  
2. *Please suggest any improvements that could be made to this subject.*
  - Should have more help/time for final exam
  - More feedback on assignments
  - More clear lectures (worked examples)
  - Change lecturer
  - Assignment 1 should be group only
  - The number of assignments should be cut down. Marks should be allocated for inspections.
  - What is expected from this subject was vague and was often clarified only during the course of doing assignments, not well before the assignments are due. A clear guideline on learning objectives is crucial.
  - We need better theoretical model (I think, I can't read the last word—JohnR)
  - Some of the material is too basic.
  - Order on the discussion board.
  - Automatic HD's all around
  - Better feedback on assignments so that we know what we did wrong. Don't make the forum *that* important.

- Too many deliverables (assignments and inspections)
- Please have much clearer marking criteria. Please be more objective in marking.
- Please provide more material earlier in the semester. Gives students more idea on the level of expectation.
- No suggestions

## 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.