Coursework Specification
Software Architecture, Process and Management

Stuart Anderson

27 January 2017

The coursework task is to develop a case study of software architecture that supports SAPM course by providing an exemplar of the use of Software Architecture. The process is designed to give you the opportunity to have some formative feedback and for you to provide feedback to your fellow students. The idea is to develop a coordinated set of web pages that describe a particular system at an architectural level together with a description of the key requirements of the system and how the software architecture makes it possible to check these requirements are fulfilled by the system

The work is organised in groups. Each group is responsible for developing a case study of their own choice of system and architecture. There is a list of candidate systems and/or architectures available on the SAPM wiki. The group list will be circulated to the SAPM mailing list. Although the work is organised into groups the assessment is predominantly individual. The assessment scheme is outlined at the end of this document. A fully detailed assessment scheme will be issued shortly.

The goal of this practical is to generate a case study of the architecture of a deployed system that will help deepen our understanding of software architecture and its role in process and management of large-scale software systems. We are around 120 people so we need to be disciplined in the way we approach this task. In particular, if everyone generated 2000 words of text we would have 240,000 words which is way too much to be useful without some sort of map. So we need to be reasonably terse, ideally 500-1000 words per person. We can assume people will not read all of the material in the wiki but that all the material will be read by some class members.

The group list will be circulated by email. When this is done you should do the following:

1. The first named person in the group should email all the group and set up a meeting to get the group to meet. I suggest you use a doodle poll for this and provide a wide range of possible times. Once you have all but two or three of the group agreed on time you should hold the meeting at an agreed time.

2. Log in to the wiki, edit your profile and add your name to the list of names on the page for your group.

3. At the first group meeting you should:

   (a) Decide who will be the **secretary** for the group. This person should take short notes of group meetings and should maintain a group page on the wiki. In particular this should record who has agreed to do something by what date so the group can keep track of commitments.

   (b) Discuss the case study you would like to carry out. Ideally people should have looked at the possible candidates listed on the SAPM wiki beforehand. You can also decide on a different system if you want to, there is no need to select from the list. Useful tips for selecting a case study:

i. If there is a system people are familiar with (for example they have worked with it in employment or in some other context). This means you have access to expertise in the group that could be helpful.
ii. Ideally you should all be interested in the system. It might be useful to get everyone in the group to indicate which ones they are interested in.
iii. At the first meeting you might not be able to decide on a single system for the case study so you may have two or three acandidates and get people to have a look at what information is available to help inform a final decision at a second meeting.

Each group will be responsible for organising the page for one case study. Each person in the group will be responsible for a page linked off their group's case study page. That page counts as their contribution (the page can refer to other pages if necessary). Contributions should always be supported by evidence wherever possible so you should make reference to literature or other evidence to support views expressed on a page. The precise structure of a case study could vary considerably but here is a list of possible sections in a case study description (not all of these need to be included in a case study):

Review Pages: Every group will have an internal and an external reviewer and each will maintain a page that records their reviewing activity and the response of the authors to review comments.

Internal Review (mandatory): This is a mandatory contribution every group must have such a page. This should be a record of the review process focussing on the internal coherence of the group's work. This should record issues that were identified and how they were resolved by the group members. The individual responsible for this page is the internal reviewer for the group.

External Review (mandatory): This is a mandatory contribution every group must have such a page. This should be a record of the review process focussing on the external linkage of the group's work to the work of other groups. This should record issues that were identified and how they were resolved by the group members. The individual responsible for this page is the external reviewer for the group.

Description of the System: This should be a high level description of the use and function of the system you have chosen.

The Software Architecture of the System: This could include diagrams documenting the prescriptive structure of the system together with an overview of what the architecture is intended to achieve. The diagrams might consider the static, dynamic and deployment structures of the system.

Discussion of the Architectural Elements used in the System: What sorts of components make up the architecture, what kinds of connectors are used in the different aspects of the architecture. What properties do the components and connectors have and why have they been selected for this system.

Key Quality Attributes: What Quality Attributes are important for the system (e.g. reliability, performance, scalability, safety, ...). There will probably be two, or three or four key attributes. This section should describe the Quality Attributes and consider why they are important for the system and how failing to achieve the desired level of Quality might affect the ability of the system to operate successfully.

Quality Attribute in detail: Here there might be one section like this for each key attribute and there should be a more detailed description of the Quality Attribute and how the software architecture facilitates controlling this Quality Attribute.

Descriptive versus Prescriptive Architecture: The descriptive architecture often omits aspect to simplify the presentation of the software architecture. This section should investigate what is omitted from the Descriptive architecture and why the omissions are important to the operation of the system.

Evolution of the Software Architecture: If the system is long lived then this section could cover the history of the system and its software architecture identifying what has driven change in the system and how this has impacted the software architecture.

Evaluation of the Success of the Software Architecture: This should be an evaluation of how successful you think the architecture has been in supporting the operation of the system. Here you should consider what evidence of success and failure you can find and evaluate the strengths and weaknesses of the system.

Process used to Develop and Operate the System: Here you should consider the development process used for the system and how this is influenced and supports the Software Architecture.

The role of the System in its Operating Organisation: In this section you should consider how the system embeds in the organisation and the extent to which its architecture influences and is influenced by the organisation it is supporting.

Review process (mandatory): This is a mandatory contribution every group must have such a page. This should document the decisions taken by the group in terms of who is doing what (that may evolve because people change their mind. It should also specify the review processes to be followed by the group in terms of ensuring the internal consistency of the work of the group (i.e. that the pages generated by the group are coherent and cross refer to one another appropriately) and in terms of ensuring the work of the group makes appropriate reference to the work of other groups and other external resources. This should be written up by an individual but it should record the deliberations of the group in agreeing the processes. This will be used to assess the group component of the final grade.

Alternative Approaches: A summary of an alternative approach to the software architecture of the system.

Exam Questions: A small number of exam style questions on the software architecture.

Summary and Map of the Case Study: An overall summary and map of the case study generated by the group.

Experience: Summarising some experience you have had in developing or maintaining software using the software architecture .

Other Ideas: If you have other ideas let me know and I'll OK them provided they are relevant.

Each group should get together, in week 3, decide on the roughly what each person is going to contribute – see the group page. Each group should also decide on their reviewing process and who will do the internal and external review activities for the group.

Activities

The main activities to be undertaken during the course:

1. Contribute multiple choice questions about your Case Study via Peerwise (15% of final grade). See my email on how to join Peerwise.

2. Comment on other people's contributions. Here you should make constructive comments that identify strong points and if there is some weakness you should make an actionable suggestion for improvement (20% of final grade). Ideally this should involve commenting inside your group and outside your group.

3. Participation in your group to work out the overall structure of your *Case Study* page and to decide on the internal and external review processes (15% of final grade).

4. Making your contribution to the wiki you have agreed to make in discussion with your group (50% of final grade).

You should spend around 25-30 hours developing your contribution and making comments.

Timeline

The timeline for the practical is as follows:

1. **Formative Feedback Deadline: 2 March 1600:** This is the deadline for the first draft of all of your contributions – peerwise questions, comments and contributions to the pages. I will use the final grading scheme to grade your work and make some suggestion on how to improve your work. This intermediate assessment is purely formative. It points to how you might improve but does not count towards your score. I will look at everything you have generated on or before 2 March.

2. **Final Deadline: 30 March 1600:** At this point I will use the final grading scheme to grade your submissions .

Submitting the coursework takes place when you submit the post to the blog there is no other submission system. Material added to the Wiki after the deadline will be ignored.

Guidelines

Your principal audience is your fellow students. You are aiming to help class members understand the system you have chosen and its Software Architecture so: your contributions should be accurate, informative, interesting and relevant; comments should focus on how to improve the material you are commenting on; peerwise questions should help people learn and retain material on your case study. To expand a little more on commenting, a comment could:

- relate the piece to something you have done (e.g. in a job, in your System Design Project, in your individual project), or read (in another piece of reading for this course, or anywhere else), or heard in lectures (in any course); and/or

- express surprise or disagreement at something in the piece, and explain carefully what the foundation of your feeling is; and/or

- write about something in the piece that you initially found hard to understand, but managed to work out, perhaps you feel that others would benefit from a discussion of your initial confusion and subsequent realisation; and/or

- discuss something in the piece that you find ambiguous or contentious, explaining carefully what the two (or more) possible readings are and inviting your readers to comment; and/or

- explore consequences of something argued in the piece, explaining carefully why certain things are consequences, and discuss whether these consequences are consistent with what you know.

A **fundamental principle** is always to be polite, always focus on the content of material, avoid personal comments or anything that might be hurtful to the reader.

Length

For your contributions, the length that you are aiming for is somewhere between 500 and 1000 words — longer is not necessarily better. If you need more words that is OK but there is a law of diminishing returns - the longer the post, the less likely anyone will ever get to the end.

Evaluation and Grading

Contributions, comments and peerwise questions will be evaluated on the basis of four equally weighted factors:

Relevance: how relevant is the contribution, comment or peerwise question to the content of the course?

Understanding: how well you demonstrate good understanding of the topic of the topic your are referring to.

Quality: how well structured is your contribution, comment or peerwise question? How clear is your work?

Impact: how accurate is your contribution, comment or peerwise question, how much bearing does it have on practice.

Overall 25 marks will be awarded for the coursework. The proportions of marks for different aspects of the practical are given above. Detailed guidance on the evaluation approach will be distributed soon. The Wiki will be sufficiently populated so you can start work on Monday 30 January.

Questions

If you have any questions regarding the coursework please post it as a comment on the SAPM wiki page and I will respond. The collected questions will be on the clarification tab on the course webpage.