



# **Board of Studies**

## **Course Descriptor Update**

**COURSE TITLE: Software Architecture Process and Management (Level 11) (INFR11038)**

**PROPOSER(S): Stuart Anderson**

**DATE: 9 Feb 2016**

## **SECTION 2 – COURSE DESCRIPTOR**

### **2d. Summary Course Description:**

*[Provide a brief official description of the course, **around 100 words**. This should be worded in a student-friendly way, it is the part of the descriptor a student is most likely to read.]*

Large-scale systems differ from small-scale system in the challenges they pose in capturing, and managing the process of delivering systems that conform to, quality attribute requirements (e.g. availability, performance, security). Software Architecture provides a framework for managing these aspects of systems. This course introduces Software Architecture concepts and provides experience of the use of Software Architecture in the development process and management of systems.

### **Course Description:**

*[Provide an academic description, an outline of the content covered by the course and a description of the learning experience students can expect to get. See guidance notes at: [http://www.studentsystems.ed.ac.uk/Staff/Support/User\\_Guides/CCAM/CCAM\\_Information\\_Captured.html#AcademicDescription](http://www.studentsystems.ed.ac.uk/Staff/Support/User_Guides/CCAM/CCAM_Information_Captured.html#AcademicDescription)].*

**Software Architecture:** Quality Attributes; Large-scale systems; Quality Attribute Scenarios; Architectural Tactics; Design Checklists; Tradeoffs; Architectural Patterns; Product Lines; Exemplar Quality Attributes (Availability; Performance; Security; Testability; Usability; ...); Case Studies (Apache, Linux, Real-time Architecture, Medical Systems, Banking Systems, ...).

**Software Architecture in the Development Process:** Architecture in Processes (Traditional – V-model, Risk-centric, Agile), Process Improvement, Modern Developments (e.g. DevOps)

Management Aspects: Tools through the process; planning; tracking; cost management; people and teams

Relevant QAA Computing Curriculum Sections: Professionalism, Software Engineering, Systems Analysis and Design

### **2e. Summary of Intended Learning Outcomes (MAXIMUM OF 5):**

*[List the learning outcomes of the course, emphasising what the impact of the course will be on an individual who successfully completes it, rather than the activity that will lead to this outcome. Further guidance is available from <https://canvas.instructure.com/courses/801386/files/24062695>]*

On completion of this course, the student will be able to

1. Integrate knowledge of Software Architecture to capture Quality Attribute requirements for a system, evaluate proposed architectures against these requirements and synthesise options for improvement.
2. Analyse and justify complex tradeoff decisions between competing software architectures.
3. Evaluate the strengths and weaknesses of software architecture in support of particular approaches to design, process and management for a particular system and make recommendations on the choice of process for that system.
4. Working in a group to critically reflect on aspects of Software Architecture literature and practice to create a resource that support their learning in Software Architecture.

## Reading List

### Required:

- Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (Third.). Addison-Wesley Professional.

### Recommended:

- Bass, L., Weber, I., & Zhu, L. (2015). *DevOps: A Software Architect's Perspective* (1st ed.). Addison-Wesley Professional.
- Bachmann, F., Bass, L., Clements, P., Garlan, D., Ivers, J., Little, M., ... Stafford, J. (2010). *Documenting Software Architectures: Views and Beyond* (Second.). Addison-Wesley Professional.
- George, F. (2010). *Just enough software architecture: a risk-driven approach*. Marshall & Brainerd.

## Assessment Information

*[Provide a description of all types of assessment that will be used in the course (e.g. written exam, oral presentation, essay, programming practical, etc) and how each of them will assess the intended learning outcomes listed above. Where coursework involves group work, it is important to remember that every student has to be assessed individually for their contribution to any jointly produced piece of work. Please include any minimum requirements for assessment components e.g. student must pass all individual pieces of assessment as well as course overall].*

- **One two-hour examination** will cover the first three learning outcomes by testing the student's capacity to solve small examples of these types of issue in exam format. The examination will be open book.
- **One 25-hour group exercise.** Students will work in groups to develop a collective asset in support of their learning of SAPM. Each group will prepare supplementary material covering a defined aspect of the course based on their reading for the course. Each student will be responsible for one page of the contribution and the group will also define a review process to help assure the quality of their contributions. Around 15% of the final grade will derive from the quality of the review process. The remainder will be assessed individually on the basis of an individual's contribution to the group deliverable and to their comments on other student's work. There will be an interim deadline (week 6) where formative feedback is provided on the work done so far. This will be provided in the form of comments and a meeting with the group to provide overall feedback (approx. one hour).

## Time spend on assignments:

*[Weightings up to a 70/30 split between exam and coursework are considered standard, any higher coursework percentage requires a specific justification. The general expectation is that a 10-point course will have an 80/20 split and include the equivalent of one 20-hour coursework assignment (although this can be split into several smaller pieces of coursework).*

*The Practical Examination category should be used for courses with programming exams. You should not expect that during term time a student will have more than 2-4 hours to spend on a single assignment for a course per week. Please note that it is possible, and in many cases desirable, to include formative assignments which are not formally assessed but submitted for feedback, often in combination with peer assessment.]*

You should expect to spend approximately 25 hours on the coursework for this course.

### **Breakdown of Learning and Teaching Activities:**

*[Total number of lecture hours and tutorial hours, with hours for coursework assignments.]*

*[The breakdown of learning and teaching activities should only include contact hours with the students; everything else should be accounted for in the Directed Learning and Independent Learning hours.*

*The total being 10 x course credits. Assume 10 weeks of lectures slots and 10 weeks of tutorials, though not all of these need to be filled with actual contact hours. As a guideline, if a 10-pt course has 20 lecture slots in principle, around 15 of these should be filled with examinable material; the rest should be used for guest lectures, revision sessions, introductions to assignments, etc. Additional categories of learning and teaching activities are available, a full list can be found at:*

[http://www.euclid.ed.ac.uk/Staff/Support/User\\_Guides/CCAM/Teaching\\_Learning.htm](http://www.euclid.ed.ac.uk/Staff/Support/User_Guides/CCAM/Teaching_Learning.htm)

Lecture Hours: 20 hours

Programme Level Learning and Teaching: 2 hours

Summative assessment hours: 2 hours

Feedback/Feedforward hours: 1 hour

Directed Learning and Independent Learning hours: 75 hours

Total hours: 100 hours

You may also find the guidance on 'Total Contact Teaching Hours' and 'Examination & Assessment Information' at:

[http://www.studentsystems.ed.ac.uk/Staff/Support/User\\_Guides/CCAM/CCAM\\_Information\\_Captured.html](http://www.studentsystems.ed.ac.uk/Staff/Support/User_Guides/CCAM/CCAM_Information_Captured.html)