SUBJECT OUTLINE

48433 Software Architecture

Course area	UTS: Engineering
Delivery	Spring 2019; City
Credit points	6cp
Requisite(s)	48440 Software Engineering Practice OR 31469 Object-oriented Design OR 31279 Software Development and Processes OR 31244 Applications Programming OR 48024 Applications Programming
Result type	Grade and marks

Recommended studies: Students are expected to have knowledge of object-oriented design and unified modelling language (UML) – this can be demonstrated through completion of any of the prerequisite subjects or the Foundations of Programming: Object-Oriented Design course available on lynda.com through the UTS: Library.A quiz is provided to check mastery of the expected prerequisite knowledge. Students are advised to attempt this quiz in the first few weeks of the session.

Subject coordinator

Dr Luke Mathieson
Email: luke.mathieson@uts.edu.au

Note: The subject coordinator may be contacted by email or phone if you have matters of a personal nature to discuss, e.g., illness, study problems, team problems, or a request for an appointment outside the given consultation hours.

Students are encouraged to learn from their peers in this subject, so please seek help from others in this subject first.

Subject description

This subject teaches students current industry practices to design, develop and evaluate software architecture meeting predefined quality characteristics of functionality (suitability, security), usability (operability), efficiency (time behaviour, resource utilisation) and maintainability (changeability, testability). Concepts, theories and technologies underlying the methods and techniques are introduced and explained as required. Students apply the industry practices that they have learned to develop an architecture of a business system.

Subject learning objectives (SLOs)

Upon successful completion of this subject students should be able to:

1. Describe the factors that affect the architectural context and requirements, including stakeholders and their interests, architectural qualities, business requirements and usage scenarios.
2. Develop and refine multiple views of a software system architecture, based on the conceptual, execution and implementation architecture.
3. Understand the key issues in choosing and implementing architectural patterns, including performance, testing, security, usability, maintainability and reliability qualities.
4. Reason about alternative architectural designs to satisfy a system's quality attributes.
5. Develop a complete software architecture for a proposed solution to a realistic industry problem.

Course intended learning outcomes (CILOs)

This subject also contributes specifically to the development of the following faculty Course Intended Learning Outcomes (CILOs) and Engineers Australia (EA) Stage 1 competencies:

- Needs, context and systems - engineering takes place within the wider context of society and the environment, which encompasses social, economic and sustainability needs (A.0)
- Problem-solving and design - engineering practice focuses on problem solving and design where artifacts are

conceived, created, used, modified, maintained and retired (B.0)

- Abstraction and modelling - Abstraction, modelling, simulation and visualisation inform decision-making, and are underpinned by mathematics, as well as basic discipline sciences (C.0)
- Evaluate model applicability, accuracy and limitations, which is linked to EA Stage 1 Competencies: 2.1, 2.2 (C.3)
- Communicate effectively in ways appropriate to the discipline, audience and purpose, which is linked to EA Stage 1 Competency: 3.2 (E.1)

Teaching and learning strategies

This subject is taught through facilitated self-learning. All lectures and related material are available online and may be viewed at the student's convenience. Quizzes are provided to enable the student test their understanding and application of knowledge about software architecture.

Students are expected to have studied the tutorial topics in preparation for tutorials. During the first half of the semester the tutorials will demonstrate the topic through application to a problem, followed by student evaluation of the application. Then students will perform an exercise and evaluate each other's work in order to develop their skills in collegial review and ability to identify where an architecture might fail.

During the second half of the semester students will be working on their major assignment. During tutorials each team will present to another team the architecture stage they have developed. The reviewing team will evaluate the presented architecture stage to acknowledge what has been done well, and where the proposed might fail. The review feedback will be communicated to the presenting team as soon as possible after the presentation.

Content (topics)

- Architecture in context
- Conceptual, implementation and execution architectures
- Architectural quality attributes
- Architectural patterns and styles
- Architecture evaluation
- Documenting architectures
- Software architecture for specific needs, for example
    - Software architecture for product lines
    - Software architecture for real time systems
    - Software architecture for web based systems

Program

Week/Session	Dates	Description
1	22 Jul	**Introduction to this subject:** Subject objectives. How the subject is taught. Topics in software architecture. Assessment, open book exam. How to do well. **Notes:** The tutorial (non-compulsory) will match the video.
2	29 Jul	Tutorial: • The architecture context • User stories • Usage narratives • Use cases • Prioritizing requirements

3	5 Aug	Tutorial: ● Conceptual architecture ● Modelling the business problem ● Evaluating the conceptual architecture
4	12 Aug	Tutorial: ● Execution Architecture ● Modelling the computer based solution ● Evaluating the execution architecture
5	19 Aug	Tutorial: ● Implementation architecture ● Evaluating the implementation architecture.
6	26 Aug	Tutorial: ● Architecture evaluation
8	2 Sept	Tutorial: ● Architecture evaluation **Notes:** Begin major assignment. From teams. Choose case study.
VC	9 Sept	Review week. No tutorial
9	16 Sept	Begin major project. **Notes:** Architecture context
10	23 Sept	**Peer review** **Notes:** Conceptual architecture.

11	30 Sept	**Peer review** **Notes:** Execution architecture
12	7 Oct	**Peer review** **Notes:** Implementation architecture
13	14 Oct	**Peer review** **Notes:** Overall project. Major assignment report due.
14	21 Oct	**Review week - no class**

Additional information

Additional information is available on UTSOnline

Assessment

In this subject while the main concpets are being learned, students are expected to develop their skills of assessing and providing feedback on the work of their peers, as they will be expected to do in the workplace.

Once the basic concepts have been learnt, students work in groups to apply those concepts to develop the architecture of a system. This will be assessed by the staff.

Assessment task 1: Required topics quiz

Intent: This assessment task encourages students to engage and learn the subject progressively. To engage in tutorial discussions effectively students need to demonstrate their understanding of the topic by answering a quiz on the relevant topic.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2 and 4

This assessment task contributes to the development of the following course intended learning outcomes (CILOs):

A.0 and B.0

Type: Quiz/test

Groupwork: Individual

Weight: 15%

Task: Watch the lecture, complete the quiz. Lectures and related material are provided online so that students can study the material at a time that suits them. A quiz is provided for each topic to enable students to check their progress toward mastering the topic.

Due: Each quiz is due before the start of the tutorial for its topic.

Criteria: Demonstrate comprehension of the topic.

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Comprehend the topic	100	1, 2, 4	A.0, B.0

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Assessment task 2: Elective topic quiz

Intent: This assessment task encourages students to study, understand and apply the means of developing specialised software architectures.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2, 3 and 4

This assessment task contributes to the development of the following course intended learning outcomes (CILOs):

B.0 and C.0

Type: Quiz/test

Groupwork: Individual

Weight: 10%

Task: There are four specialised software architectures suited to particular problems. Choose any two of them, study them, then answer their respective quizzes.

Due: Week 12
The chosen quizzes must be completed by the end of semester, i.e. 11:59 PM, Friday October 18.

Criteria: Knowledge and understanding of the selected topic

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Understand and explain the topic	100	1, 2, 3, 4	B.0, C.0

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Assessment task 3: Team project

Intent: Work in a team to develop an architecture for a specific system in a given context.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2, 3, 4 and 5

This assessment task contributes to the development of the following course intended learning outcomes (CILOs):

A.0, B.0 and E.1

Type: Project

Groupwork: Group, group and individually assessed

Weight: 35%

Task: The project is a group assignment to be completed in small teams. You are expected to develop a software architecture for a non-trivial system using the techniques learned in this subject.

Due: Week 12
The final report must be submitted no later than 11:59 PM, Friday 18 October.

Criteria: The architectural context reflects an understanding of the problem and its stakeholders.

The developed architecture provides the required functionality and meets its quality objectives.

Rationales for architectural decisions and preferred alternatives are credible.

An individual reflective journal provides the weekly entries throughout the assignment to record your reflection on the learning journey.

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Architectural context	10	1	A.0
Developed architecture	60	2, 5	B.0
Architectural rationale	10	3	B.0, E.1
Individual reflective journal	20	4	E.1

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Further information: The pool of marks for each group will be communicated to each group who will then allocate the marks within the group to reflect the relative contribution of each group member to the result. The resultant allocation must be returned to the subject coordinator in time to be included in final mark calculations. Every effort will be made to complete marking this assessment item and its mark communication to each team within one week of its submission.

Assessment task 4: Peer review

Intent: Develop and practice the professional skill of reviewing and critiquing the work of peers.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

4

This assessment task contributes to the development of the following course intended learning outcomes (CILOs):

B.0 and E.1

Type: Laboratory/practical

Groupwork: Group, individually assessed

Weight: 15%

Task: Each group will present their work in progress to another group for their review and critique. Reviews are completed using a review form (currently Google forms) and submitted on completion. Peer reviews are conducted in the tutorial room.

Due: Peer reviews will be completed during the tutorial for each week of the major project. Reviews must be completed on the same day of the presentation.

Criteria: Observations and suggestions are well informed and impartial.

Observations and suggestions are intended to improve the work.

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Relevant and informative	50	4	B.0
Useful Improvements	50	4	B.0, E.1

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Assessment task 5: Final examination

Intent: Assess a student's knowledge of software architecture and to apply that knowledge to a given problem.

Objective(s): This assessment task addresses the following subject learning objectives (SLOs):

1, 2 and 3

This assessment task contributes to the development of the following course intended learning outcomes (CILOs):

B.0 and C.3

Type: Examination

Groupwork: Individual

Weight: 25%

Task: Demonstrate your knowledge of software and ability to apply it to a range of problems in both general and specialised domains. The final exam will contain both multiple-choice questions and subjective scenario-based descriptive questions.

No supplementary exam will be given in this subject. If you miss any assessment through documented illness or misadventure, you should consult the subject coordinator.

Due: UTS Exam period
This will be an open book exam.

Criteria: Demonstrate adequate knowledge of software architecture.

Solve a given software architecture problem.

Evaluate a given architecture in its context.

Apply software architecture principles in a specialised domain.

Criteria linkages:

Criteria	Weight (%)	SLOs	CILOs
Demonstrate knowledge	25	2	B.0
Problem solving	25	2	B.0
Solution evaluation	25	3	C.3
Specialised domains	25	1	B.0

SLOs: subject learning objectives
CILOs: course intended learning outcomes

Assessment feedback

Feedback for the required and elective topics is provided through quizzes that provide multiple attempts and can be taken at any time prior to any tutorials involving the topic. During tutorials feedback is frequent and ongoing through discussions and responses to questions. During formal assignments feedback is provided through peer reviews that examine specific stages and aspects of the assignment. When the major assignment is completed the overall assignment is assessed and feedback is provided within two weeks.

Minimum requirements

In order to pass the subject, a student must achieve an overall mark of 50% or more.

Prize offered

WiseTech Global Software Engineering Prize

Recommended texts

Any of the following texts are recommended for this subject:

Bass, L., Clements, P. and Kazman, R. (2003), Software architecture in practice, Addison-Wesley, 2nd ed or 3rd ed

Gorton, I. (2006), Essential Software Architecture, Springer, Berlin

Reekie, J. and McAdam, R. (2006), A Software Architecture Primer, Angophora Press, Sydney

References

The following books are all in the UTS library. You may wish to use these to further your knowledge of the field.

Bosch, J. (2000), Design and use of software architectures : adopting and evolving a product-line approach, Harlow, Addison-Wesley

A nice introduction to the “industrial" view of software architecture.

Bass, L., Clements, P. and Kazman, R. (2003), Software architecture in practice, Addison-Wesley, 2nd ed.

Useful supplementary reading for students that want to read a lot.

Hofmeister, C., Nord, R. and Soni, D. (2000), Applied software architecture, Addison- Wesley, Reading, Mass

Interesting look at factors and system context.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. and Stal, M. (1996), Pattern-Oriented Software Architecture: A system of patterns, John Wiley & Sons, New York

Contains a good collection of the basic architectural patterns.

Brown, W. J., Malveau, R. C. and Mowbray, T. J. (1998), AntiPatterns : refactoring software, architectures, and projects in crisis, Wiley, New York

Excellent. Contains patterns at all levels, not just architecture.

Clements, P., Bachmann, F., Bass, L. and Garlan, D. (2002), Documenting software architecture: views and beyond, Addison-Wesley, Boston

A slightly different set of view types than in this subject.

Gorton, I. (2006), Essential Software Architecture, Springer, Berlin

Good book from a local Australian author.

Other resources

Lectures

All lectures have been recorded and are available in video format through UTSOnline. Students enrolled in this subject are expected review these lectures at their convenience.

Announcements during the semester will be posted on UTSOnline, http://online.uts.edu.au/. All students are automatically registered with UTSOnline. It is your responsibility to check UTSOnline on a regular basis. Any queries will be replied to by the subject coordinator within the week. Students are free to reply to other students' questions or comments. Please show your fellow students and staff an appropriate level of courtesy when corresponding.

Courseware
Tutorial exercises and study modules for each foundation module 1 through 6 will be posted to UTSOnline during the semester.

Information on writing style (including ‘writing better English’), developing an argument and a referencing guide can be found at http://www.lib.uts.edu.au/information/writing_skills

Standards

ISO 42000 series of standards for software and systems architecture describe current best practice. These do not specify a method for developing an architecture but do specify the requirements for an architecture description. A standard for architecture evaluation is also being developed.

Graduate attribute development

For a full list of the faculty's graduate attributes and EA Stage 1 competencies, refer to the FEIT Graduate Attributes webpage.

Assessment: faculty procedures and advice

Extensions

When, due to extenuating circumstances, you are unable to submit or present an assessment task on time, please contact your subject coordinator before the assessment task is due to discuss an extension. Extensions may be granted up to a maximum of 5 days (120 hours). In all cases you should have extensions confirmed in writing.

Special Consideration

If you believe your performance in an assessment item or exam has been adversely affected by circumstances beyond your control, such as a serious illness, loss or bereavement, hardship, trauma, or exceptional employment demands, you may be eligible to apply for Special Consideration.

Late Penalty

Work submitted late without an approved extension is subject to a late penalty of 10 per cent of the total available marks deducted per calendar day that the assessment is overdue (e.g. if an assignment is out of 40 marks, and is submitted (up to) 24 hours after the deadline without an extension, the student will have four marks deducted from their awarded mark). Work submitted after five calendar days is not accepted and a mark of zero is awarded.

For some assessment tasks a late penalty may not be appropriate – these are clearly indicated in the subject outline. Such assessments receive a mark of zero if not completed by/on the specified date. Examples include:

weekly online tests or laboratory work worth a small proportion of the subject mark, or
online quizzes where answers are released to students on completion, or

professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date, or
take-home papers that are assessed during a defined time period, or
pass/fail assessment tasks.

Querying marks/grades and Final Results

If a student disagrees with a mark or a final result awarded by a marker:

where a student wishes to query a mark, the deadline for a query during teaching weeks is 10 working days from the date of the return of the task to the student

where a student wishes to query a final examination result, the deadline is 10 working days from the official release of the final subject result.

Further information can be found at Academic advice.

Academic liaison officer

Academic liaison officers (ALOs) are academic staff in each faculty who assist students experiencing difficulties in their studies due to: disability and/or an ongoing health condition; carer responsibilities (e.g. being a primary carer for small children or a family member with a disability); and pregnancy.

ALOs are responsible for approving adjustments to assessment arrangements for students in these categories. Students who require adjustments due to disability and/or an ongoing health condition are requested to discuss their situation with an accessibility consultant at the Accessibility Service before speaking to the relevant ALO.

The ALO for undergraduate students is:

Chris Wong
telephone +61 2 9514 4501

Statement about assessment procedures and advice

This subject outline must be read in conjunction with the policy and procedures for the assessment for coursework subjects.

Statement on copyright

Teaching materials and resources provided to you at UTS are protected by copyright. You are not permitted to re-use these for commercial purposes (including in kind benefit or gain) without permission of the copyright owner. Improper or illegal use of teaching materials may lead to prosecution for copyright infringement.

Statement on plagiarism

Plagiarism and academic integrity

At UTS, plagiarism is defined in Rule 16.2.1(4) as: 'taking and using someone else's ideas or manner of expressing them and passing them off as ... [their] own by failing to give appropriate acknowledgement of the source to seek to gain an advantage by unfair means'.

The definition infers that if a source is appropriately referenced, the student's work will meet the required academic standard. Plagiarism is a literary or an intellectual theft and is unacceptable both academically and professionally. It can take a number of forms including but not limited to:

- copying any section of text, no matter how brief, from a book, journal, article or other written source without duly acknowledging the source
- copying any map, diagram, table or figure without duly acknowledging the source
- paraphrasing or otherwise using the ideas of another author without duly acknowledging the source
- re-using sections of verbatim text without using quote marks to indicate the text was copied from the source (even if a reference is given).

Other breaches of academic integrity that constitute cheating include but are not limited to:

- submitting work that is not a student's own, copying from another student, recycling another student's work, recycling previously submitted work, and working with another student in the same cohort in a manner that exceeds the boundaries of legitimate cooperation
- purchasing an assignment from a website and submitting it as original work

- requesting or paying someone else to write original work, such as an assignment, essay or computer program, and submitting it as original work.

Students who condone plagiarism and other breaches of academic integrity by allowing their work to be copied are also subject to student misconduct Rules.

Where proven, plagiarism and other breaches of misconduct are penalised in accordance with UTS Student Rules Section 16 – Student misconduct and appeals.

Avoiding plagiarism is one of the main reasons why the Faculty of Engineering and IT is insistent on the thorough and appropriate referencing of all written work. Students may seek assistance regarding appropriate referencing through UTS: HELPS.

Work submitted electronically may be subject to similarity detection software. Student work must be submitted in a format able to be assessed by the software (e.g. doc, pdf (text files), rtf, html).

Further information about avoiding plagiarism at UTS is available.

Retention of student work

The University reserves the right to retain the original or one copy of any work executed and/or submitted by a student as part of the course including, but not limited to, drawings, models, designs, plans and specifications, essays, programs, reports and theses, for any of the purposes designated in Student Rule 3.9.2. Such retention is not to affect any copyright or other intellectual property right that may exist in the student's work. Copies of student work may be retained for a period of up to five years for course accreditation purposes. Students are advised to contact their subject coordinator if they do not consent to the University retaining a copy of their work.

Statement on UTS email account

Email from the University to a student will only be sent to the student's UTS email address. Email sent from a student to the University must be sent from the student's UTS email address. University staff will not respond to email from any other email accounts for currently enrolled students.