## Contents

1 Overview .......................................................... 1
   1.1 Description .................................................. 1
   1.2 Outcomes .................................................. 2
       1.2.1 Career outcomes ......................................... 2
       1.2.2 Course outcomes ......................................... 2
   1.3 Prerequisites ................................................. 2
   1.4 Module credits ............................................... 3
   1.5 Related modules ............................................. 3

2 Contact details of instructors and support staff .......... 3
   2.1 Course coordinator and lecturer .......................... 3
   2.2 Assistant lecturers ......................................... 3
   2.3 Administrative support ..................................... 3

3 Study material .................................................. 4
   3.1 Mandatory study material .................................. 4
   3.2 Optional study material .................................... 4

4 Organisation ..................................................... 4
   4.1 Lectures ..................................................... 4
   4.2 Module web site ............................................ 4
   4.3 Announcements .............................................. 4
   4.4 Communication with instructors ............................. 5

5 Schedule .......................................................... 5
   5.1 Lecture schedule ............................................ 5
   5.2 Schedule for mini-project .................................. 6
   5.3 Schedule for main project .................................. 6
   5.4 Tests and exams ............................................. 6

6 Assessment ........................................................ 7
   6.1 Plagiarism policy ............................................ 7
   6.2 Semester tests ............................................... 7
   6.3 Exam .......................................................... 7
   6.4 Mini projects ................................................. 7
   6.5 Main project ................................................ 7
   6.6 Year mark calculation ...................................... 8

7 Epilogue .......................................................... 8
1 Overview

- COS-301 is meant to be an exciting course which exposes you to
  - challenges,
  - principles,
  - theory,
  - methods, and
  - tools
commonly encountered in industrial-size software-development projects.

1.1 Description

Software Engineering aims to introduce engineering principles into software development. The aims of the engineering principles are to

- ensure that the resultant system satisfies both, functional and non-functional requirements,
- have predictable outcomes and hence repeatability,
- use design metrics, preferably with defined construction tolerances,
- use mathematical models where applicable,
- ensure that the system is failure tolerant,
- achieve component reuse,
- separate design from implementation, and
- have control over quality making appropriate quality attribute trade-offs.

An engineering approach requires

- professionalism, including
  - adherence to a code of conduct, and potentially
  - legal accountability,
- adherence to standards and methods,
- sensible division of work across roles with defined
  - responsibilities and
  - skills requirements,
- solid quality assurance with the option of proving correctness and completeness mathematically, and
- solid tool support for all aspects of the development and maintenance processes.

In this course, students are exposed to most of these engineering principles (with the exclusion of formal methods) and acquire the skills typically required of a software engineer.
1.2 Outcomes

This module exposes students to challenges encountered in software development projects of industrial scale, applying sound software engineering principles and practices.

1.2.1 Career outcomes

Candidates who completed this course will have gained an understanding of the concepts and principals of software engineering. They should be able to work in a professional software development team and should be able to guide the team on best practices and software development processes.

Students will have an understanding of required engineering practices and software engineering tools used for software development, quality assurance, configuration management, and project management.

1.2.2 Course outcomes

Skills acquired in this module include

- the ability to effective interface and manage a client,
- to be able to plan and execute a larger software development project,
- able to develop software within a modular architecture requiring the integration of modules developed by different teams,
- to be able to work within a software development team with specialized roles and responsibilities,
- to understand and be able to select and use appropriate software development methodologies,
- to be able to use software engineering tools including design tools, development tools, build tools, testing tools, and configuration management tools,
- to perform the requirements elicitation, documentation and validation of a realistic software development project,
- to design an architecture and the functionality for a software product,
- to introduce and enforce quality assurance practices, and
- to provide suitable documentation for a software development project,

1.3 Prerequisites

The prerequisite modules are COS-110 and COS121.
1.4 Module credits

The module is based on the ACM/IEEE Computing Curriculum for Computer Science with emphasis on Software Engineering, covering many of the topics of the *Guide to the Software Engineering Body of Knowledge* (SWEBOK) and the SCM/IEEE Guidelines for Undergraduate Degree Programs in Software Engineering.

The module credits are 27 based on 2 lectures and 1 practical per week as a year module.

1.5 Related modules

This module relates to most undergraduate modules in that this module will lead you to practically apply the knowledge and skills acquired during your undergraduate studies.

This is the first, introductory course in software engineering. This course is complemented and rounded off by the following honours-level modules:

- **COS 730: Software Engineering I**: This aims to deepen the understanding of requirements engineering, software development methodologies, quality assurance and covers component based development in depth.

- **COS 731: Software Engineering II (Software Architecture)**: This course focuses on software architecture. The main aim is to enable software architects to define, validate, document and analyze software architectures which aim to provide an infrastructure within which non-functional requirements like performance, scalability, security, integrability/accessibility, reliability, auditability, testability, and cost can be appropriately addressed. The course covers architectural patterns and tactics, reference architectures, architecture design methodologies, architecture description languages and how to document software architectures as well as architecture analysis methods.

2 Contact details of instructors and support staff

2.1 Course coordinator and lecturer

Name: Dr. Fritz Solms  
Office: IT 4-38  
Email: fsolms@cs.up.ac.za  
Telephone: 012 420 2547

2.2 Assistant lecturers

Name: Stacey Omeleze  
Email: staceyaomeleze@gmail.com  
Cell: 072 937 4870

Name: Lecton Ramasila  
Email: lectonlm@gmail.com

2.3 Administrative support

Academic administrator: Elmarie Willemse  
Office: IT 4-18  
Telephone number: 012 420 2547  
Email address: ewillemse@cs.up.ac.za
3 Study material

The following study material applies to this course:

3.1 Mandatory study material

All mandatory study material will be made available through the course portal. There is no prescribed text book for this course.

3.2 Optional study material

The following study material is optional. It may assist in deepening the students understanding of various aspects of the course.


2. *OMG’s Unified Modeling Language Superstructure*, version 2.4.1:
   - URL: http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/

3. The *Guide to the Software Engineering Body of Knowledge* (SWEBOK)
   - URL: http://www.swebok.org

4 Organisation

This section discusses the practical way in which the course is conducted.

4.1 Lectures

The lecture times and venues are published in the University Handbook. Note that it is recommended that students prepare for lectures.

4.2 Module web site


4.3 Announcements

Announcements will be made regularly through the course portal. It is important that students regularly (at least weekly) visit the course portal to check for new announcements and new material which is made available. Note that the department reserves the right to deviate from planned schedules under circumstances which cannot be foreseen and that some announcements and/or instructions on the module web site might be of critical and mandatory nature.
4.4 Communication with instructors

Students are encouraged to ask questions in class during lectures. This enables other students to also benefit from the discussion and may assist in addressing questions and concerns when they are pertinent.

If your questions or concerns are not fully addressed in class, you may contact the lecturer per email to either raise the question in the email or to make an appointment to see the lecturer.

5 Schedule

The main course components are

- the lectures covering the theory,
- a mini-project which aims to prepare you for the main project,
- a main project, and
- formal tests and exams.

The schedule for each of the above is listed in separate tables.

5.1 Lecture schedule

The planned lectures for the first quarter are as follows:

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/2</td>
<td>Intro to Software Engineering, Version control</td>
</tr>
<tr>
<td>6/2</td>
<td>Scoping, functional requirements, data structure design</td>
</tr>
<tr>
<td>9/2</td>
<td>Data structure design and mini project overview</td>
</tr>
<tr>
<td>16/2</td>
<td>Services Contracts &amp; process design</td>
</tr>
<tr>
<td>20/2</td>
<td>Technology and architecture neutral analysis and design</td>
</tr>
<tr>
<td>23/2</td>
<td>Architecture Design 1</td>
</tr>
<tr>
<td>27/2</td>
<td>Architecture Design 2</td>
</tr>
<tr>
<td>2/3</td>
<td>Maven &amp; Java-EE reference architecture 1</td>
</tr>
<tr>
<td>6/3</td>
<td>Java-EE reference architecture 2</td>
</tr>
<tr>
<td>9/3</td>
<td>Dependency Injection</td>
</tr>
<tr>
<td>16/3</td>
<td>Unit testing</td>
</tr>
<tr>
<td>20/3</td>
<td>Continuous Integration and Integration Testing</td>
</tr>
<tr>
<td>13/4</td>
<td>Usability principles and testing</td>
</tr>
<tr>
<td>20/4</td>
<td>Development processes</td>
</tr>
</tbody>
</table>

The remaining lectures will cover further topics in software engineering like aspect-oriented programming, estimation, CMMI, metrics, design principles, software support, bug tracking and maintenance as well as a range of other interesting software engineering topics.

The second semester students will be exposed also to some industry presentations which aim to convey typical challenges, skills requirements, opportunities and practices used in industry.
## 5.2 Schedule for mini-project

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>9/2</td>
<td>Team Allocation – Requirements and Application Design phase</td>
</tr>
<tr>
<td>13/2</td>
<td>Client requirements session</td>
</tr>
<tr>
<td>27/2</td>
<td>Submission of Requirements and Application design</td>
</tr>
<tr>
<td>27/2</td>
<td>Team Allocation – Architecture Design phase</td>
</tr>
<tr>
<td>2/3</td>
<td>Master requirements and application design provided</td>
</tr>
<tr>
<td>9/3</td>
<td>Architecture Specification</td>
</tr>
<tr>
<td>9/3</td>
<td>Team Allocation – Implementation and Integration phase</td>
</tr>
<tr>
<td>13/3</td>
<td>Master architecture specification provided</td>
</tr>
<tr>
<td>13/3</td>
<td>Meetings with teams to discuss modules</td>
</tr>
<tr>
<td>27/3</td>
<td>Mid-Implementation Demo</td>
</tr>
<tr>
<td>17/4</td>
<td>Post-Implementation Demo</td>
</tr>
<tr>
<td>17/4</td>
<td>Testing phase teams allocated</td>
</tr>
<tr>
<td>24/4</td>
<td>Final Demo</td>
</tr>
</tbody>
</table>

## 5.3 Schedule for main project

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>13/2</td>
<td>Client’s invited to submit project proposals</td>
</tr>
<tr>
<td>13/3</td>
<td>Client’s project proposal’s received</td>
</tr>
<tr>
<td>14/4</td>
<td>Project proposals refined and assessed</td>
</tr>
<tr>
<td>24/4</td>
<td>Projects list provided to students</td>
</tr>
<tr>
<td>4/5</td>
<td>Project proposals submitted by 7h00</td>
</tr>
<tr>
<td>4/5</td>
<td>Project tenders submitted to clients</td>
</tr>
<tr>
<td>7/5</td>
<td>Client team selections received</td>
</tr>
<tr>
<td>8/5</td>
<td>Main projects allocated</td>
</tr>
<tr>
<td>22/5</td>
<td>Demo 1 (Slots a)</td>
</tr>
<tr>
<td>29/5</td>
<td>Demo 1 (Slots b) and documents submission</td>
</tr>
<tr>
<td>24/7</td>
<td>Demo 2 (Slots a)</td>
</tr>
<tr>
<td>31/7</td>
<td>Demo 2 (Slots b) and documents submission</td>
</tr>
<tr>
<td>21/8</td>
<td>Demo 3 (Slots a)</td>
</tr>
<tr>
<td>28/8</td>
<td>Demo 3 (Slots b) and documents submission</td>
</tr>
<tr>
<td>18/9</td>
<td>Demo 4 (Slots a)</td>
</tr>
<tr>
<td>22/9</td>
<td>Demo 4 (Slots b) and documents submission</td>
</tr>
<tr>
<td>16/10</td>
<td>Demo 5 and documents submission</td>
</tr>
<tr>
<td>23/10</td>
<td>Project Day (tentative)</td>
</tr>
</tbody>
</table>

## 5.4 Tests and exams

- **Test 1:** 11 May
- **Exam:** During June exam session
- **Test 2:** 21 September
6 Assessment

The course uses a continuous assessment approach with a large number of measures taken throughout the years contributing the final mark. It is hence important that students work consistently throughout the year in order to pass this course.

6.1 Plagiarism policy

This department considers plagiarism as a serious offense. Disciplinary action will be taken against student who commit plagiarism. For a formal definition of plagiarism, the student is referred to http://www.ais.up.ac.za/plagiarism/index.htm
(From the UP Main page follow the Library quick link and then the Plagiarism link)

6.2 Semester tests

Students will write one semester test per semester. Each semester test contributes 10% to the final mark for the course.

6.3 Exam

The exam will be written during the June examination session. The exam will contribute 20% to the final mark for the course.

6.4 Mini projects

The purpose of the mini-project is to a variety of teams dynamics and multi-team projects and to prepare you for the main project. The assessment of the mini-project will be mainly based on peer reviews, assessment of repository contributions, demos and feedback received from the assistant lecturers. The mini project contributes 15% to the final mark.

6.5 Main project

The main project makes up 45% of your final mark. It will be continuously assessed through assessment of produced artifacts, peer review, presentations and the assessments done on project day including CS staff and client assessment.

The main project will be assessed as follows:

• 30% project demo day.
• 70% continuous assessment made up of
  – quality of the code and tests as assessed off the version control system,
  – quality of the demos and presentations,
  – quality of the documentation,
  – an assessment from the client, and
  – an individual contribution mark reflecting the relative work done by individual group members based on git repository histories and peer reviews.
6.6 Year mark calculation

The year mark will be calculated via

\[ Y = 0.1T_1 + 0.1T_2 + 0.2E + 0.15P_{\text{mini}} + 0.45P_{\text{main}} \]

7 Epilogue

The COS-301 teaching team and the Department of Computer Science would like to wish you success in this Software Engineering Module. We hope that you will find it insightful, interesting, challenging and that the insights and skills you acquired during this module will be valuable for you throughout your professional life.