COURSE DETAILS
Course Code: F27IP
Full Course Title: Industrial Project: Software Development Methods
SCQF Level: 7
SCAF Credits: 15
Available as Elective: No

DELIVERY LEVEL
Undergraduate: Yes
Postgraduate Taught: No
Postgraduate Research: No

COURSE AIMS
This is an Industrial Project course that consists of two parts: Work-based Learning (WBL) and an Industrial Project.

The WBL part of the course will deliver work-based blended on-line learning material. Topics learned may be subject to continuous e-assessment. This course introduces topics in legacy systems, software development life cycle, system configuration and release management.

The Industrial Project part of the course will require the student to perform/implement an industrial project, embedded in and contextualised for the host company, focusing on the practical techniques learned in the WBL part of the course.

LEARNING OUTCOMES – SUBJECT MASTERY

• Understand legacy systems' architectures and technologies; and be able to assess the risks towards modernisation (e.g. towards virtualisation)
• Understand high level activities in the Software Development Life Cycle, Software Deployment and Maintenance
• Understand methods, technologies and tools of software release management and version control
• To be able to use a software version control tools to manage software development, e.g. Git
• To be able to use virtualisation technologies for software development and deployment, e.g. Docker
• To be able to scope, plan, implement and evaluate an industrial project
• To understand the time and effort involved in implementing an industrially-based project

LEARNING OUTCOMES – PERSONAL ABILITIES

• To be able to relate learned knowledge to work place computing projects
• To be aware of distinctive features of work-based blended learning.
• To be able to identify, define, and analyse alternative project scenarios
• Take some responsibility for their work and for a range of resources

• To be able to communicate effectively with colleagues at work place and extract and organise requirements effectively
### SYLLABUS

- Modern software architecture and design principles: object-oriented software design; model-based design; design patterns

- Legacy systems: software architectures and technologies; risk associated with replacing or keeping legacy systems; common pitfalls and mitigations

- Software Development Life Cycle (SDLC): software Deployment; Software Maintenance; Continuous Integration (e.g. Jenkins)

- System configuration and Release Management: system configuration process; Version control (e.g. Git); Software release management methods (e.g. itSMF, ITIL); Software release management tools (e.g. Puppet, Plutora); Virtual Machine (e.g. Docker, VirtualBox) usage for development and deployment

- Industrial project development (incl. project scope and design, planning, requirements engineering, system implementation, software testing and evaluation, and critical assessment)

### COURSE RELATIONSHIPS

N/A

### LOCATION AND ASSESSMENT METHODS

<table>
<thead>
<tr>
<th>Edi</th>
<th>SBC</th>
<th>Ork</th>
<th>Dub</th>
<th>Malay</th>
<th>IDL</th>
<th>COLL</th>
<th>ALP</th>
<th>OTH</th>
<th>Method</th>
<th>Weight</th>
<th>Exam Mins</th>
<th>Type</th>
<th>Diet</th>
<th>Synoptic Course</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coursework</td>
<td>100</td>
<td></td>
<td>Assessment</td>
<td>Semester 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Coursework</td>
<td>100</td>
<td></td>
<td>Reassessment</td>
<td>Semester 1</td>
<td></td>
</tr>
</tbody>
</table>