Software engineering project

Outcomes

A student:

- justifies methods used to plan, develop and engineer software solutions **SE-12-01**
- applies structural elements to develop programming code **SE-12-02**
- analyses how current hardware, software and emerging technologies influence the development of software engineering solutions **SE-12-03**
- evaluates practices to safely and securely collect, use and store data **SE-12-04**
- explains the social, ethical and legal implications of software engineering on the individual, society and the environment **SE-12-05**
- justifies the selection and use of tools and resources to design, develop, manage and evaluate software **SE-12-06**
- designs, develops and implements safe and secure programming solutions **SE-12-07**
- tests and evaluates language structures to refine code **SE-12-08**
- applies methods to manage and document the development of a software project **SE-12-09**

Content

Identifying and defining

- Define and analyse the requirements of a problem

  Including:
  - demonstrating need(s) or opportunities
  - assessing scheduling and financial feasibility
  - generating requirements including functionality and performance
  - defining data structures and data types
  - defining boundaries

- Explore tools used to develop ideas and generate solutions

  Including:
  - brainstorming, mind-mapping and storyboards
  - data dictionaries, including selecting appropriate data types
  - algorithm design
  - code generation
  - testing and debugging
  - installation
  - maintenance

- Investigate types of software implementation methods

  Including:
  - direct
  - phased
  - parallel
  - pilot

Research and planning

- Research and use the Waterfall software development approach

  Including:
  - logical progression of steps used throughout the life cycle
  - stages of ‘falling water’
  - advantages and disadvantages
  - scale and types of developments

- Research and use the Agile software development approach

  Including:
  - rate of developing a final solution
  - method tailoring
  - iteration workflow
  - scale and types of developments

- Research the WAgile software development approach

  Including:
  - understanding it is a hybrid model
  - analysis of the ‘when’ and ‘how’ intervention is applied during the development life cycle
  - scale and types of developments

- Apply project management to plan and conduct the development and implementation of a project and software engineering solution

  Including:
  - scheduling and tracking using a software tool, including Gantt charts
  - using collaboration tools

- Explore social and ethical issues associated with project work, including working individually, collaboratively and responding to stakeholders
- Explore communication issues associated with project work

  Including:
  - involving and empowering the client
  - enabling feedback
  - negotiating

- Investigate how software engineering solutions are quality assured

  Including:
  - defining criteria on which quality will be judged
  - ensuring requirements are met using a continual checking process
  - addressing compliance and legislative requirements

- Demonstrate the use of modelling tools

- Explain the contribution of back-end engineering to the success and ease of software development

  Including:
  - technology used
  - error handling
  - interfacing with front end
  - security engineering

Producing and implementing

- Design, construct and implement a solution to a software problem using appropriate development approach(es)
- Present a software engineering solution using presentation software
- Develop, construct and document algorithms
- Allocate resources to support the development of a software engineering solution
- Demonstrate the use of programmed data backup
- Implement version control when developing a software engineering solution
- Explore strategies to respond to difficulties when developing a software engineering solution

  Including:
  - looking for a solution online
  - collaboration with peers
  - outsourcing
- Propose an additional innovative solution using a prototype and user interface (UI) design

Testing and evaluating

- Apply methodologies to test and evaluate code
- Use a language-dependent code optimisation technique
- Analyse and respond to feedback
- Evaluate the effectiveness of a software engineering solution

  Including:
  - developing a report to synthesise feedback
  - developing a test plan
  - testing data used/generated based on path and boundary testing
  - comparing actual output with expected output