**Scheme of work - Software Development**

**Lifecycles**

|  |  |  |  |
| --- | --- | --- | --- |
| Academic Term | Fall 2022 | **HTU Credit Value** | 3 |
| Programme Title | Higher Nationals in Computing | Lecture Duration | 2 |
| HTU Course Title | Software Development Lifecycles | Lab Duration | 2 |
| HTU Course No. | 040201220 | Classroom | S207 |
|  |  |  |  |
| BTEC Unit Title | Software Development Lifecycles | BTEC Credit Value | 15 |
| BTEC Unit No, | 7 | GLH | 60 |
| BTEC Unit Code | K/618/7408 | ULH | 90 |
| BTEC Unit Level | 5 | TQT | 150 |

**Class Schedule:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Section No | Sunday | Monday | Tuesday | Wednesday | Thursday |
| Hamzah Alkofahi | 1 | 08:30 am – 10:30 am |  | 08:30 am – 10:30 am |  |  |
| Hamzah Alkofahi | 2 | 10:30 am – 12:30 pm |  | 10:30 am – 12:30 pm |  |  |
| Ashraf Smadi | 3 | 12:30 pm – 12:30 pm |  | 12:30 pm – 12:30 pm |  |  |
| Asma’a Lafi | 4 |  | 08:30 am – 10:30 am |  | 08:30 am – 10:30 am |  |
| Salem Alemaishat | 5 |  | 10:30 am – 12:30 pm |  | 10:30 am – 12:30 pm |  |

**Note: Asma?a Shukry Yousef Lafi Hana' Zaid Abdul Karim Al Rashid**

Learners should spend lesson time and non-supervised time working on assignments.

**Course Description:**

The unit introduces students to lifecycle decision making at different stages of the

software development process. They will examine various lifecycle models and learn

to appreciate their particular characteristics in order to understand for which project

environments they are most appropriate. Theoretical understanding will be translated

into practical skills through an actual software development lifecycle project. Students

will become confident in the use of particular tools and techniques relevant to a

chosen methodology.

|  |  |
| --- | --- |
| **Learning Outcomes (LO)** | **Assessment 1** |
| **LO1 Describe** different software development lifecycles |  |
| **LO2** Explain the importance of a feasibility study |  |
| **LO3** Undertake a software development lifecycle |  |
| **L04** Discuss the suitability of software behavioral  design techniques |  |

| Sessions | Learning Outcome(s) | Session Activities |
| --- | --- | --- |
| Session 1 | LO1  Topic: SDLC models | Sample activities:   * Group discussion on the systems development lifecycle (SDLC) and the value of this expertise to the organisation. Explain use of predictive (Waterfall, Prototyping, RAD) and adaptive (Spiral, Agile, DSDM) software development models. * Group activity: Finding examples of SDLC projects to introduce a new product or service. * Show video of a predictive and adaptive SDLC case study. * Introduce SDLC as a scientific approach to analysis and synthesis activities in a systems development. * Activity: Discuss suitability of different SDLC models   depending on the development environment. |
| Session 2 | LO1  Topic: SDLC stages | Sample activities:   * Discussion on lifecycle stage and connectivity: (feasibility study, analysis, design, implementation, testing, review) or (analysis, design, implementation, maintenance, planning). * Group activity: Discussion to explore how SDLC activities can   be organised effectively.   * Show video of a case study, describing the activities and linkage between them. * Activity: Discuss ways of tracing the system requirements throughout the SDLC. * Activity: Students to review their own knowledge by identifying ways of handling changes in an SDLC. |
| Session 3 | LO1  Topic: Testing | Sample activities:   * Activity: Discuss the importance of developing test harnesses, black box/white box testing and incremental testing in the test phase of an SDLC. Use case studies and scenarios to demonstrate phased build of test environments appropriate to the testing strategy. * Group activity: Working in groups discuss the benefits/drawbacks of implementing a risk based test strategy. |
| Session 4 | LO1  Topic: Integration | Sample activities:   * Identify and provide examples of software integration and its impact on building out test environments. * Discuss the importance of a version control procedure to manage software releases. * Group activity: Make a scenario where a software release is not accepted by the customer and suggest potential consequences and solutions. |
| Session 5 | LO1  Topic: Changeover | Sample activities:   * Discuss the importance of changeover strategies, procedures and processes in relation to trials and Go-Live prerequisites. * Activity: Compare the risks associated with different   changeover strategies including phased, pilot, parallel and direct.   * Explore the importance of contingency planning as part of a changeover strategy and provide examples. * Group activity: Identify and discuss the major issues in changeover to operational Go-Live in a computing project. |
| Session 6 | LO2  Topic: Analysis – software requirement | Sample activities:   * Discuss requirement gathering techniques, for example: interviews, observation, investigation of documentation, questionnaires or focus groups. * Activity: Engage students in interview role playing to   establish requirements. Use the scenario to derive complete, concise and unambiguous requirements.   * Group activity: Students roleplay to observe a task and record it. Afterwards compare notes to show subjective differences. |
| Session 7 | LO2  Topic: Analysis – key drivers | Sample activities:   * Explore key drivers for SDLC (e.g. performance and efficiency, legacy systems upgrade, automation, elimination of human error). * Discuss the importance of identifying key drivers to establish   scope of SDLC project.   * Introduce students to the concept of AS-IS / TO-BE system state as explain using an example. * Activity: Students use a case study to extract key drivers. * Group activity: Discuss the importance of deriving key drivers to justify carrying out the SDLC. |
| Session 8 | LO2  Topic: Analysis – feasibility study | Sample activities:   * Discuss the need for a feasibility study: Identifying legal, social, economic, technical and timescales factors. * Group activity: Discuss SDLC purpose and anticipated outcomes in relation to proposed solutions. * Activity: Students to make a comparative analysis between two competing solutions. |
| Session 9 | LO2  Topic: Analysis – types of requirements | Sample activities:   * Introduce the notion of functional and non-functional user requirements, and constraints. * Discuss how MosCoW method can be used to prioritise requirements. * Activity: Students to be given a case study to investigate the MosCoW prioritisation of requirements and types. |
| Session 10 | LO2  Topic: Analysis – summary | Sample activities:   * Discuss, using case study examples, the importance of clarifying requirements using sorting and filtering of collected data. * Activity: Using a case study example, complete a team activity that demonstrates the importance of requirements gathering as an iterative process. |
| Session 11 | LO3  Topic: SDLC execution – scoping | Sample activities:   * Introduce students to stakeholders, requirements identification and specification (e.g. scope, inputs, outputs, processes and process descriptors). * Discuss the importance of selecting the right solution   aligned with technology and business roadmaps. Discuss quality assurance required.   * Activity: Group discussions. Topics can be selected by each   group, researched and debated by each team member. The focus should be the challenge of scoping an SDLC project agreeable to the client and supplier. |
| Session 12 | LO3  Topic: SDLC execution – constraints | Sample activities:   * Constraints: Specific to activity (e.g. costs, organisational policies, legacy systems, hardware requirements). * Group discussion of constraints specific to activity (e.g. costs, organisational policies, legacy systems, hardware requirements). * Discuss how constraints impact on selection of solution for development. * Group activity: Research how SDLC projects use trade-off negotiations to agree constraints. |
| Session 13 | LO3  Topic: SDLC execution – documentation | Sample activities:   * Discuss the importance of developing report documentation throughout the SDLC (including background information, problem statements, data collection process and summary, recommendations, appendices) and its impact on successful project outcomes. * Activity: Students to investigate the importance of the need for documentation to be produced and benefits from requirements traceability throughout the SDLC. |
| Session 14 | LO3  Topic: SDLC execution – design | Sample activities:   * Systems analyst guest speaker to discuss the importance of systems analysis terminology and tools in design phase (including data stores and entities, data flows; process representation techniques, data relationships – 1:1, 1:Many (1:M) and Many: Many (M:M)). * Activity: Students to ask questions to speaker. |
| Session 15 | LO3  Topic: SDLC execution – design | Sample activities:   * Introduce students to design and engineering techniques relevant to chosen methodology (e.g. Context Diagrams, Data Flow Diagrams (DFDs), Entity Relationship Diagrams (ERDs), Business Systems Options (BSOs), Technical Systems Options (TSOs)) and quality considerations (e.g. Total Quality Management (TQM)). * Group activity: Students to investigate the importance of HCI   in the design of solutions. |
| Session 16 | LO4  Topic: Formal behavioural design techniques | Sample activities:   * Techniques: Flowcharts, pseudocode and formal specification methods. * Introduce students to software behavioural design techniques including flowcharts, pseudocode and formal specification methods. * Activity: Produce an analysis of the use of formal specification techniques to determine software behaviour. * Group activity: Discuss the importance of statistical techniques to evaluate the performance behaviour of software. |
| Session 17 | LO4  Topic: Event/state/data driven behavioural design techniques | Sample activities:   * Introduce event/state/data driven techniques to specify software behaviour. Discuss, using an example, the advantages of each technique and where it is appropriate. * Activity: Discuss the advantages of using data driven design   techniques in maintaining software.   * Group activity: Discuss the use of state driven software for particular applications and sketch corresponding FSM state diagrams. |
| Session 18 | LO4  Topic: Finite state machine behavioural design technique | Sample activities:   * Introduce, using an example, the FSM technique to design software behaviour. * Discussion on use of e-FSM to describe system states of cooperating software (e.g. telephone network). * Introduce the notion of state explosion for e-FSMs. * Group activity: Select an application and construct a state diagram to specify the behaviour of the control software. |
| Session 19 | LO4  Topic: Reachability, safety and liveness | * Introduce, using examples the notion of reachability, safety and liveness in software. * Activity: Research a control application and investigate techniques to test for reachability, safety and liveness properties in the software. * Group activity: Assess the implications of behavioural design errors in safety critical software. |
| Session 20 | LO4  Topic: Automatic analysis and animation tools | * Introduce state of the art software analysis tools to measure and test for reachability, safety and liveness properties. * Activity: Demonstrate the use of animation tools to exercise software. * Group activity: Survey and make comparative analysis of   latest automatic analysis and animation tools. |

|  |  |  |
| --- | --- | --- |
|  |  |  |