**Unit 7: Software Development Life Cycles**

**ASSIGNMENT BRIEF PART 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **Pearson BTEC Level 5 Higher National Diploma in Computing** | | |
| **Unit number** | Unit 7: Software Development LifeCycles | | |
| **Assignment title** | Undertake a Software Development Lifecycle | | |
| **Academic Year** |  | | |
| **Unit Tutor** |  | | |
| **Issue date** |  | **Submission date** |  |

|  |
| --- |
| **Submission Format:** |
| *Format:*   * You must use the Times font with 12pt size, turn on page numbering; set line spacing to 1.3 and margins to be as follows: left = 1.25cm, right = 1cm, top = 1cm, bottom = 1cm. Citation and references must follow the Harvard referencing style. * The submission is in the form of the following:   **1. A System Implementation and Evaluation Report**  A formal report documenting the implementation of the software development lifecycle (SDLC) for your chosen project scenario. The report must include:   * A detailed investigation into business needs and stakeholder requirements. * Analysis of software requirements and their traceability throughout the lifecycle. * Evaluation of the systems investigation process and its effectiveness in improving software quality. * Discussion of software behavioural design techniques used during implementation.   The recommended word limit is 2,000–2,500 words, although you will not be penalised for exceeding the total word limit.  You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced using the Harvard referencing system.  **2. A formal 10-minute presentation (10–20 slides as a guide, with speaker notes)** Prepare a formal presentation to communicate your findings to a technical audience. The presentation should highlight:   * Key outcomes from the systems investigation process. * Effectiveness of your approach in improving software quality. * The behavioural design techniques selected and their suitability for the project. * Justifications for how data-driven approaches can enhance software reliability.   The presentation should consist of:   * **10–20 slides** as a guide. * Speaker notes accompanying each slide to provide context for your discussion.   You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced using the Harvard referencing system.  **3. A Prototype for a Small Application Feature** with at least two functionalities of the choosen scenario.  The prototype should demonstrate at least **two core functionalities** from **two different modules** within the proposed system. It must serve as a practical demonstration during your final **presentation and live demo**, aiming to convincingly address the stated requirements.   * The prototype should align with the system requirements and behavioural design principles outlined in your report. * It must be clearly structured and functional enough to showcase how the solution meets both user needs and technical expectations. * The prototype must be **compressed into a .zip file** and **submitted via the CMS** platform as part of your assessment submission.   Assessment will consider the prototype's relevance, clarity, and effectiveness in supporting your system implementation and evaluation.  *Submission:*   * Students are compulsory to submit the assignment in due date and in a way requested by the Tutor. * The form of submission will be a soft copy posted on <http://cms.btec.edu.vn/>.   *Note:*   * The individual Assignment must be your own work, and not copied by or from another student. * If you use ideas, quotes or data (*such as diagrams*) from books, journals or other sources, you must reference your sources, using the Harvard style. * Make sure that you understand and follow the guidelines to avoid plagiarism. Failure to comply this requirement will result in a failed assignment. |
| **Unit Learning Outcomes:** |
| **LO3** Undertake a software development lifecycle  **LO4** Discuss the suitability of software behavioral design techniques. |
| **Transferable skills and competencies developed** |
| **Transferable Skills and Competencies Developed**  The completion of the **System Implementation and Evaluation Report** will enable students to develop a wide range of transferable skills and competencies. These skills are essential for academic success and employability in the software development industry. They align with the learning outcomes (LO3 and LO4) and are directly applicable to real-world projects.  **Skills Developed**  **1. Problem-Solving Skills**  Students will enhance their ability to:   * Investigate business needs and identify relevant stakeholder requirements. * Apply critical thinking to analyze complex software development challenges during implementation. * Develop innovative solutions for non-routine problems encountered throughout the lifecycle.   **2. Technical Skills**  The assignment will equip students with:   * Competence in using software analysis tools such as Context Diagrams, ERDs, DFDs, BSOs, and TSOs. * Proficiency in tracing requirements throughout the software lifecycle using traceability matrices. * Expertise in applying behavioural design techniques such as FSMs, extended FSMs, flowcharts, pseudocode, and formal specification methods. * Understanding of data-driven approaches to improve software reliability and scalability.   **3. Analytical Skills**  Through systems investigation and evaluation, students will develop:   * The ability to critically evaluate the effectiveness of systems investigations in improving software quality. * Proficiency in analyzing behavioural tools and techniques for suitability in specific project environments. * Capacity to assess the impact of data-driven software on reliability and effectiveness.   **4. Communication Skills**  Students will learn to:   * Articulate technical concepts clearly in written reports that document implementation processes and evaluations. * Present findings effectively to technical audiences via structured presentations with speaker notes.   **Competencies Developed**  **1. Research Skills**  The assignment requires students to conduct thorough research on SDLC methodologies, behavioural design techniques, and quality improvement approaches. This fosters active learning, methodical investigation, and evidence-based decision-making.  **2. Professional Documentation**  By producing a comprehensive System Implementation and Evaluation Report, students will develop professional writing skills essential for communicating technical information effectively in workplace settings.  **3. Decision-Making**  Students will practice making informed decisions about:   * Lifecycle model selection based on project constraints. * Software quality improvement approaches tailored to specific needs. * Behavioural design techniques suitable for achieving project goals.   **4. Adaptability**  Students will demonstrate adaptability by:   * Responding effectively to changes in stakeholder requirements or project scope. * Implementing solutions that address challenges encountered during systems investigation.   **Relevance to Employability**  The skills developed through this assignment are directly applicable to roles such as:   1. Software Developer 2. Systems Analyst 3. Quality Assurance Engineer 4. Project Manager 5. Business Analyst   These roles require proficiency in implementing software development lifecycles, evaluating system effectiveness, analyzing behavioural design techniques, and improving software quality—competencies central to this assignment. |
| **Vocational scenario** |
| **Vocational Scenario**  You are working as a **Software Development Team Lead** at **TechVision Solutions**, a leading software engineering firm specializing in delivering customized software solutions for various industries. After successfully completing the planning phase in the form of a **Software Development Plan (SDP)**, you are now tasked with implementing and evaluating the next phase of the project. This phase focuses on conducting a detailed systems investigation, implementing key stages of the software development lifecycle (SDLC), and critically evaluating the effectiveness of your approach.  In this role, you are responsible for producing a **System Implementation and Evaluation Report** to document your findings and processes. Additionally, you will deliver a **formal presentation** to communicate your results to the technical team and stakeholders. These deliverables will demonstrate your ability to undertake a complete SDLC while evaluating the suitability of behavioural design techniques and improving software quality.  **Project Scenarios**  You have been assigned one of the following project scenarios to implement and evaluate:   1. **Hotel Room Booking System**   A chain of hotels requires a centralized web-based system to streamline their operations and improve customer experience. The system must include:   * Booking Management: Allow customers to book rooms online with real-time availability updates. * Payment Processing: Integrate secure payment gateways supporting multiple methods such as credit cards, e-wallets, and bank transfers. * Customer Profiles: Enable customers to create accounts, manage personal information, and view booking history. * Reporting Tools: Generate reports on occupancy rates, revenue trends, and customer demographics.  1. **Food Delivery**   A startup aims to launch a web-based food delivery platform with real-time tracking and order management capabilities. The app must include:   * **Order Placement:** Allow users to browse restaurant menus, place orders, and customize preferences. * **Delivery Tracking:** Provide real-time GPS tracking for delivery personnel. * **Payment Integration:** Support secure online payments via multiple methods. * **Restaurant Management Dashboard:** Enable restaurant owners to update menus, manage orders, and analyze sales data.  1. **Learning Management System (LMS)**   A university seeks a web-based Learning Management System (LMS) for managing academic activities. The LMS must include:   * **Course Management:** Organize courses, upload materials, and manage student enrollment efficiently. * **Student Interaction:** Facilitate communication between students and instructors through forums or messaging systems. * **Assessment Tools:** Enable instructors to create quizzes, assignments, and track student progress. * **Analytics Dashboard:** Provide insights into student performance and course effectiveness.   **4. Hospital Management Software**  A hospital requires a web-based solution to enhance operational efficiency and patient care. The system must include:   * **Patient Records Management:** Securely store patient information, including medical history and treatment plans. * **Doctor Scheduling:** Organize doctor appointments and availability efficiently. * **Treatment Workflow Tracking:** Monitor treatment plans, medication schedules, diagnostic results, and progress updates. * **Billing System:** Automate invoicing and payment tracking for patient.   **Your Responsibilities**  As part of this role-play scenario, you are required to:   1. Conduct a detailed systems investigation to meet the business needs outlined in your chosen scenario:    * Gather stakeholder requirements using interviews or surveys.    * Define project scope including inputs, outputs, processes, constraints, and limitations. 2. Use appropriate software analysis tools (e.g., Context Diagrams, Data Flow Diagrams (DFDs), Entity Relationship Diagrams (ERDs)) to document your findings. 3. Trace requirements throughout the lifecycle stages using traceability matrices:    * Ensure that all requirements are tracked from initial gathering through design, implementation, testing, and deployment. 4. Evaluate the effectiveness of your systems investigation in improving software quality:    * Discuss two approaches for improving software quality (e.g., Total Quality Management (TQM), automated testing). 5. Apply behavioural design techniques such as flowcharts, pseudocode, finite state machines (FSMs), or extended FSMs during implementation:    * Evaluate their suitability for your chosen project scenario. 6. Justify how data-driven approaches improve software reliability and effectiveness:    * Highlight how data-driven designs enhance scalability while reducing errors through automation.   **Deliverables**  **1. System Implementation and Evaluation Report**  You will produce a formal report that includes:   * A comprehensive investigation into business needs and stakeholder requirements. * Documentation of software analysis tools/techniques used during the lifecycle (e.g., Context Diagrams, ERDs). * An analysis of how requirements were traced throughout the lifecycle using traceability matrices. * An evaluation of the effectiveness of your systems investigation in improving software quality. * A critical discussion on behavioural design techniques (e.g., FSMs, pseudocode) and their suitability for your project scenario. * Justifications for how data-driven approaches enhance software reliability and effectiveness.   This report must be structured professionally with headings, subsections, diagrams where applicable, and supported by research referenced using the Harvard referencing style.  **2. Formal Presentation**  Prepare a professional presentation summarizing your findings from the implementation process. The presentation should:   * Highlight key outcomes from your systems investigation process. * Discuss the effectiveness of your approach in improving software quality. * Evaluate behavioural design techniques selected for your project scenario. * Justify how data-driven approaches improve system reliability and scalability.   The presentation should consist of:   * 10–20 slides as a guide. * Speaker notes accompanying each slide to provide context for your discussion.   **3. A Simple Prototype**  You will develop a functional prototype that demonstrates **at least two core features** from **two clearly distinct functional modules** within your system. These modules must represent **separate areas of functionality** — for example, **Login and Register** would be considered part of the same *Authentication* module and **do not** count as two distinct features.  The prototype will be used during your **presentation and live demo** to effectively showcase how your system meets both user and technical requirements.  The prototype should:   * Represent functional modules that are clearly separated in purpose, such as *Authentication*, *Course Management*, *User Dashboard*, or *Reporting*. * Demonstrate key functionalities aligned with the system design and stakeholder needs outlined in your report. * Be structured using appropriate design techniques and demonstrate behavioural considerations. * Be interactive enough to illustrate how different parts of the system work together in a realistic context.   **Why These Deliverables Are Required**  These deliverables reflect real-world practices in software development projects:   1. The **System Implementation and Evaluation Report** ensures that all aspects of system development are documented comprehensively while providing evidence of how stakeholder requirements are met through proper investigation and design techniques. 2. The **Formal Presentation** allows you to communicate technical findings effectively to stakeholders or technical audiences—an essential skill in professional environments. 3. The **Prototype** enables you to demonstrate the practical implementation of your system in a realistic scenario. It showcases your ability to turn design and requirements into working features across distinct system modules. This reflects the expectations of real-world software projects, where tangible proof of functionality is essential to gain stakeholder trust and validate technical decisions. |
| **Assignment Brief and Guidance:** |
| **Activities**  **Activity 3: Undertake a Software Development Lifecycle**  **3.1 Conduct a Systems Investigation**  You need to investigate the business needs of their chosen project scenario (e.g., Hotel Room Booking System, Food Delivery, LMS).  **Steps:**   1. **Identify Stakeholder Requirements**    * Gather functional requirements (e.g., booking systems, payment processing) and non-functional requirements (e.g., performance, security, scalability) using interviews, surveys, or observations.    * Document findings in structured formats. 2. **Define Project Scope**    * Specify inputs, outputs, processes, constraints, and limitations.    * Explore alternate solutions and security considerations. 3. **Use Software Analysis Tools**    * Create Context Diagrams to illustrate system boundaries and external entities.    * Develop Data Flow Diagrams (DFDs) to show how data flows through processes.    * Design Entity Relationship Diagrams (ERDs) to represent database structures.    * Evaluate Business Systems Options (BSOs) and Technical Systems Options (TSOs). 4. **Develop a Prototype for a Small Application Feature**   Implement a prototype that includes **at least two core functionalities**, each from **two different functional modules** of the system.   * + *Note:* Two features within the same module (e.g., **Login** and **Register** in the Authentication module) will be considered **one module only** and do **not** meet the requirement.   + **Example:** A login system (from the Authentication module) and a booking form for hotel rooms or food orders (from the Booking module) are considered functionalities from **different modules**.   Use suitable technologies such as:   * + **HTML/CSS/JavaScript** for front-end development,   + **MySQL** for database design and management,   + **PHP** or other server-side technologies for back-end processing.   Deploy the prototype on a **local server** (e.g., XAMPP, WAMP) or a **cloud platform** (e.g., Heroku, Firebase) to demonstrate feasibility and allow for testing during presentation.   1. **Deliverables for the Report:**    * Include diagrams (Context Diagrams, DFDs, ERDs).    * Provide supporting documentation explaining stakeholder needs and project scope.    * Include screenshots or descriptions of the implemented feature with code snippets.   **3.2 Trace Requirements Throughout the Lifecycle**  You need to analyze how stakeholder requirements are traced throughout the software lifecycle.  **Steps:**   1. **Develop a Requirements Traceability Matrix**    * Link stakeholder requirements to system features, design elements, test cases, and implemented components. 2. **Track Requirements Across Lifecycle Stages**    * Ensure traceability from initial gathering to implementation and testing phases. 3. **Document Changes in Requirements**    * Analyze their impact on project scope, timeline, and quality. 4. **Deliverables for the Report:**    * Include the traceability matrix as part of the report.    * Provide analysis of how requirements were tracked effectively.   **3.3 Apply Software Quality Approaches**  You need to discuss two approaches for improving software quality during implementation.  **Steps:**   1. **Approach 1: Manual Testing or Automation testing**    * Test core functionalities of the prototype (e.g., login process or booking form validation).    * Document test cases with expected results and actual outcomes. 2. **Approach 2: Peer Code Review**    * Conduct peer reviews of code to identify potential bugs or improvements in logic.    * Discuss how this process ensures quality standards are met. 3. **Deliverables for the Report:**    * Include a discussion of both approaches with examples from your project scenario.    * Provide metrics used to measure software quality improvement.   **3.4 Evaluate the Effectiveness of Systems Investigation**  You need to critically evaluate their systems investigation process.  **Steps:**   1. **Highlight Strengths**    * Accuracy of tools used (e.g., diagrams) or completeness of documentation. 2. **Identify Weaknesses**    * Challenges encountered during investigation (e.g., unclear stakeholder communication). 3. **Provide Recommendations**    * Suggest refinements in stakeholder communication or analysis techniques. 4. **Deliverables for the Report:**    * Include an evaluation section discussing strengths, weaknesses, and recommendations for improvement.   **Activity 4: Discuss the Suitability of Software Behavioural Design Techniques**  **4.1 Explore Behavioural Design Techniques**  You need to discuss examples of behavioural design techniques used in their project scenario.  **Steps:**   1. Identify behavioural techniques such as:    * Flowcharts for visualizing workflows and decision points.    * Pseudocode for defining algorithms or logic flows.    * Finite State Machines (FSMs) for modeling system states and transitions. 2. Provide examples applied to your chosen project scenario:    * Use diagrams or pseudocode to demonstrate their application. 3. Deliverables for the Report:    * Include examples of flowcharts, pseudocode, or FSMs applied to your project scenario.   **4.2 Analyze Behavioural Tools**  You need to analyze a range of behavioural tools/techniques used during implementation.  **Steps:**   1. Discuss tools such as:    * Event-driven designs for responding to user actions.    * State/data-driven approaches for managing complex systems.    * Formal specification methods for ensuring precision in design. 2. Deliverables for the Report:    * Include an analysis section comparing behavioural tools used in your project scenario.   **4.3 Compare FSMs and Extended FSMs**  The difference between finite state machines (FSMs) and extended FSMs.  **Steps:**   1. Highlight key features, capabilities, and limitations of FSMs vs extended FSMs. 2. Provide examples applied to your chosen project scenario:    * Use diagrams or pseudocode to demonstrate their application. 3. Deliverables for the Report:    * Include comparisons between FSMs and extended FSMs with application examples.   **4.4 Justify Data-Driven Approaches**  Yopu need to present justifications for how data-driven software improves reliability and effectiveness.  **Steps:**   1. Discuss how data-driven designs enhance:    * Scalability under increased loads.    * Adaptability to changing requirements.    * Error reduction through automation. 2. Outline strategies for integrating data-driven approaches into your system. 3. Deliverables for the Report:    * Include justifications supported by examples from your chosen project scenario.. |
| **System Implementation and Evaluation Report Template** |
| **1. Executive Summary**   * Provide a concise summary of the report, including:   + The purpose of the project.   + Key functionalities developed.   + Major findings and conclusions from the systems investigation and evaluation.   **2. Introduction**  **2.1 Purpose**   * State the purpose of the report, emphasizing its role in documenting the implementation and evaluation phases of the SDLC.   **2.2 Project Overview**   * Provide a brief description of the selected project scenario (e.g., Hotel Room Booking System, Food Delivery,… )   **2.3 Scope**   * Define the scope of the report, including objectives, deliverables, and key functionalities developed during implementation.   **3. Systems Investigation**  **3.1 Stakeholder Requirements**   * Identify functional requirements (e.g., booking systems, payment processing) and non-functional requirements (e.g., performance, scalability, security). * Document how stakeholder needs were gathered using methods such as interviews or surveys.   **3.2 Project Scope**   * Define inputs, outputs, processes, constraints, and limitations. * Explore alternate solutions and security considerations.   **3.3 Software Analysis Tools**  Include diagrams such as:   1. Context Diagrams: Illustrate system boundaries and external entities. 2. Data Flow Diagrams (DFDs): Show how data flows through processes. 3. Entity Relationship Diagrams (ERDs): Represent database structures. 4. Business Systems Options (BSOs): Explore alternative solutions. 5. Technical Systems Options (TSOs): Evaluate technical approaches.   **4. Prototype Implementation**  **4.1 Developed Features**   * Implement a small feature of the application with **at least two functionalities from two different modules**. * *Note:* Features from the same module (e.g., Login and Register under Authentication) will be counted as one. * Example: A login system (Authentication module) and a booking form (Booking module) for hotel rooms or food orders… or choose any functions in the chosen scenario. * Technologies used:   + Front-end: HTML/CSS/JavaScript…. or any framework   + Back-end: PHP, Nodejs, Python,… or any framework   + Database: SQL Server, Mysql, MongoDB,… or and framework   **4.2 Deployment**   * Deploy the prototype locally or on a cloud platform to demonstrate feasibility. * Include screenshots or links to deployed features.   **5. Requirements Traceability**  **5.1 Traceability Matrix**  **Develop a matrix linking stakeholder requirements to:**   1. System features. 2. Design elements. 3. Test cases. 4. Implemented components.   **5.2 Lifecycle Tracking**  **Document how requirements were tracked across lifecycle stages:**   1. **From initial gathering to implementation and testing phases.**   **5.3 Change Management**  **Analyze changes in requirements:**   1. Discuss their impact on project scope, timeline, and quality.   **6. Software Quality Approaches**  **6.1 Manual Testing**  Test core functionalities of the prototype manually:   1. Example: Validate login process or booking form functionality. 2. Document test cases with expected results and actual outcomes.   **6.2 Peer Code Review**  Conduct peer reviews of code:   1. Identify potential bugs or improvements in logic. 2. Discuss how this process ensures quality standards are met.   **7. Evaluation of Systems Investigation**  **7.1 Strengths**  Highlight strengths such as:   1. Accuracy of tools used (e.g., diagrams). 2. Completeness of documentation.   **7.2 Weaknesses**  Identify challenges encountered during investigation:   1. Example: Unclear stakeholder communication or technical constraints.   **7.3 Recommendations**  Suggest refinements in stakeholder communication or analysis techniques.  **8. Behavioural Design Techniques**  **8.1 Examples of Techniques**  Discuss examples applied to your project scenario:   1. Flowcharts for visualizing workflows. 2. Pseudocode for defining algorithms or logic flows. 3. Finite State Machines (FSMs) for modeling system states and transitions.   **9. Analysis of Behavioural Tools**  **9.1 Range of Tools**  Analyze tools such as:   1. Event-driven designs for responding to user actions. 2. State/data-driven approaches for managing complex systems. 3. Formal specification methods for ensuring precision in design.   **10. FSM vs Extended FSM**  **10.1 Key Differences**  Highlight features, capabilities, and limitations of FSMs vs extended FSMs.  **10.2 Application Examples**  Provide examples applied to your chosen project scenario:   1. Use diagrams or pseudocode to demonstrate their application.   **11. Data-Driven Approaches**  **11.1 Reliability and Effectiveness**  Discuss how data-driven designs enhance:   1. Scalability under increased loads. 2. Adaptability to changing requirements. 3. Error reduction through automation.   **11.2 Implementation Strategies**  Outline strategies for integrating data-driven approaches into your system. |

|  |  |  |
| --- | --- | --- |
| Learning Outcomes and Assessment Criteria | | |
| Pass | Merit | Distinction |
| **LO3** Undertake a software development lifecycle | | **D3** Evaluate the process of undertaking a systems investigation with regard to its effectiveness in improving a software quality. |
| **P5** Undertake a software investigation to meet a business need.  **P6** Use appropriate software analysis tools/techniques to carry out a software investigation and create supporting documentation. | **M3** Analyse how software requirements can be traced throughout the software lifecycle.  **M4** Discuss two approaches to improving software quality. |
| **LO4** Discuss the suitability of software behavioural design techniques | | **D4** Present justifications of how data driven software can improve the reliability and effectiveness of software. |
| **P7** Discuss, using examples, the suitability of software behavioural design techniques. | **M5** Analyse a range of software behavioural tools and techniques.  **M6** Differentiate between a finite state machine (FSM) and an extended FSM, providing an application for both. |