**Unit 7: Software Development LifeCycles**

**ASSIGNMENT BRIEF PART 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **Pearson BTEC Level 5 Higher National Diploma in Computing** | | |
| **Unit number** | Unit 7: Software Development LifeCycles | | |
| **Assignment title** | Planning 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.   **A Software Development Plan (SDP),** The submission will take the form of a comprehensive Software Development Plan (SDP) that addresses all learning outcomes and assessment criteria.  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.  *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:** |
| **LO1** Describe different software development lifecycles.  **LO2** Explain the importance of a feasibility study. |
| **Transferable skills and competencies developed** |
| The completion of the**Software Development Plan (SDP)**assignment will enable students to develop a range of transferable skills and competencies. These skills are essential for academic success and professional development in the software development field, preparing students for real-world challenges and employability in computing-related roles. **Computing-related cognitive skills**   * **Knowledge and Understanding**   + Demonstrate knowledge and understanding of essential facts, concepts, principles, and theories related to software development lifecycles.   + Use this knowledge to model and design software systems that meet specific project requirements, ensuring comprehension, communication, and prediction of trade-offs. * **Problem Analysis**   + Recognize and analyze criteria and specifications appropriate to specific problems within the software development lifecycle.   + Plan strategies for solving these problems using structured approaches. * **Evaluation**   + Critically analyze the extent to which a proposed software system meets predefined criteria for current use and future scalability.   + Assess the suitability of lifecycle models for different project environments. * **Application of Theory**   + Deploy appropriate theories, practices, and tools to design, implement, and evaluate computer-based systems effectively.   + Apply lifecycle methodologies to ensure quality assurance throughout the software development process.   **Computing-related practical skills**   * **System Evaluation**   + Evaluate software systems based on quality attributes such as performance, reliability, scalability, and usability.   + Identify trade-offs presented by constraints within the given problem domain. * **Project Planning**   + Plan and manage projects to deliver software systems within constraints of requirements, timescale, and budget.   + Develop structured project plans using tools such as Gantt charts, traceability matrices, ERDs, DFDs, etc. * **Risk Management**   + Recognize risks associated with deploying software systems in specific contexts (e.g., technical risks or legal compliance).   + Implement risk mitigation strategies while adhering to safety standards. * **Documentation**   + Effectively use tools for constructing and documenting software applications.   + Develop comprehensive Software Development Plans that include feasibility studies, lifecycle model analysis, risk management strategies, system designs, and recommendations. * **Problem-Solving**   + Critically evaluate complex problems with incomplete information.   + Devise appropriate solutions within constraints such as budget or organizational policies.   **Generic skills for employability**   * **Intellectual Skills**   + Develop critical thinking abilities to analyze lifecycle models and feasibility criteria.   + Make evidence-based arguments when selecting methodologies or proposing solutions.   + Improve numeracy skills when estimating costs or evaluating technical feasibility. * **Self-Management**   + Enhance self-awareness through reflection on project outcomes.   + Set goals and plan actions independently while adapting to changing project requirements.   + Act on initiative to innovate solutions for complex software development challenges. * **Contextual Awareness**   + Understand how software systems meet the needs of individuals, businesses, or communities.   + Recognize how workplaces are governed and adapt solutions accordingly (e.g., adhering to organizational policies or legal frameworks).   **Relevance to Employability**  The skills developed through this assignment are directly applicable to roles such as:   1. Software Developer 2. Business Analyst 3. Systems Analyst 4. Project Manager 5. Quality Assurance Engineer   These roles require proficiency in planning, analyzing, documenting, and managing software development projects—competencies central to this assignment. By completing the SDP assignment, students will gain practical experience in applying theoretical concepts to real-world scenarios while developing critical skills valued by employers in the computing industry. |
| **Vocational scenario** |
| You are a **Software Development Consultant** at **InnovateTech Solutions**, a global leader in custom software solutions. Your expertise in planning and implementing software development lifecycles has earned you the responsibility of leading the planning phase for a critical project. Your client has approached you to design and plan a robust web-based software system tailored to their specific needs.  As part of this project, you are required to produce a comprehensive **Software Development Plan (SDP)** that outlines the lifecycle methodology, feasibility analysis, functional and non-functional requirements, and system design strategies. This plan will serve as the foundation for the development process, ensuring that the system meets the client’s expectations while adhering to industry standards.  **A. Project Scenarios:**  **Choose one of the following scenarios**, each requiring a detailed SDP tailored to its unique requirements:   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   **B. Project Requirements:**   * 1. **User Requirement Analysis:**   2. Each project scenario includes specific user requirements that must be addressed in your SDP:      1. Develop a user-friendly interface with intuitive navigation tailored to the target audience (e.g., hotel staff, app users).      2. Ensure accessibility across devices (e.g., desktops, tablets, mobile phones).      3. Implement secure login systems with role-based access control to protect sensitive data.   **1.2. Functional Requirements:**   * **Front-end:**  1. The system must provide a responsive interface compatible with desktops, tablets, and mobile devices. 2. Allow users to search by relevant criteria:  * For hotels: Room availability by date or location. * For food delivery: Menu items by category or price range. * For LMS: Courses by department or instructor name.  1. Include clear navigation menus for accessing different modules (e.g., booking management or patient records).  * **Back-end:**   1. Enable administrators to manage records such as bookings, orders, or patient details.   2. Implement secure authentication systems to ensure role-based access control: * Customers can only access their profiles. * Administrators can access all data within their scope of authority.   1. Integrate with relational databases (e.g., MySQL or PostgreSQL) to store structured data securely. * **Notifications:**  1. Send automated email or SMS notifications for key events:  * Booking confirmations * Delivery status updates * Appointment reminders * **Reporting Tools:**  1. Generate detailed reports based on user activity or business performance metrics such as:  * Occupancy rates for hotels * Delivery times for food orders * Student performance in an LMS   **1.3. Infrastructure Requirements:**   * 1. **Hosting and website management:** Use web hosting services to ensure the website runs smoothly and is always available.   2. **Web server:** Use Apache or Nginx for the web server.   3. **Security:** Implement security techniques like SSL, data encryption, and protection against XSS and CSRF attacks.   **1.4. Design Requirements:**   * 1. Create a modern, simple yet visually appealing design to attract users.   2. The website should include essential components such as navigation menus, a shopping cart, and product review sections.   3. Categories should be well-organized for easy product navigation.   **1.5. Performance Requirements:**   * 1. The website must have fast load times and perform well even under heavy traffic.   2. Ensure the system is scalable for future growth and upgrades.   **C. Learning Objectives:**  Students will apply knowledge of the SDLC, from requirements gathering, analysis, design, and implementation to testing and maintenance of the system.  **D. Steps for Implementation:**   * **Requirement Analysis:** Identify both functional and non-functional requirements for the fashion e-commerce system. * **System Design:** Plan the database structure, system architecture, and user interface. * **Implementation and Testing:** Develop the necessary features and conduct testing to ensure the system functions as required. * **Maintenance:** Carry out maintenance tasks after the system is officially launched.   Students will work in teams/ Individual to complete each phase of the SDLC, learning how to manage a software project efficiently and practically. |
| **Assignment Brief and Guidance:** |
| **Activity 1: Software Development Lifecycle Models**  Develop the **Software Development Plan (SDP)** by exploring and evaluating different software development lifecycle models applicable to your chosen project scenario. Your SDP should include the following:  **1. Overview of Software Development Lifecycle Models**   * Provide an overview of **two iterative models** (e.g., Agile, Spiral) and **two sequential models** (e.g., Waterfall, V-Model). * For each model:   + Describe its key characteristics, stages, and workflows.   + Explain its suitability for specific project environments.   + Include diagrams or charts to illustrate their processes.   **2. Risk Management in SDLC Models**   * Identify potential risks associated with each lifecycle model, such as:   + Technical risks (e.g., integration challenges).   + Resource risks (e.g., budget or time constraints).   + Process risks (e.g., scope creep in iterative models). * Discuss strategies for managing these risks effectively:   + Risk identification techniques.   + Mitigation plans.   + Monitoring processes.   **3. Recommendation of an Appropriate SDLC Model**   * Recommend one lifecycle model that best suits your chosen project scenario. * Support your recommendation with:   + A discussion of the benefits and drawbacks of your chosen model.   + Judgments on its appropriateness based on project requirements, constraints, and expected outcomes.   **4. Assessment of the Waterfall Model for Large Projects**   * Assess the merits of applying the Waterfall lifecycle model to large-scale projects. * Compare its suitability against other models in terms of:   + Stability and predictability.   + Resource management.   + Scalability for complex projects.   **Activity 2: Feasibility Study**  Develop a feasibility study section within your **Software Development Plan (SDP)** that evaluates the viability of your chosen project scenario. Your feasibility study should include the following:  **1. Contents and Purpose of a Feasibility Report**  **Purpose:**   * Explain why feasibility studies are essential for software development projects. * Highlight their role in:   + Reducing risks.   + Ensuring project viability.   + Supporting decision-making processes.   **Comparison of Technical Solutions:**   * Describe methods for comparing technical solutions, such as:   + Cost-benefit analysis.   + Decision matrices. * Provide examples relevant to your chosen project scenario.   **Components of a Feasibility Report:**   * Discuss key components such as:   + Technical feasibility: Compatibility with existing systems.   + Economic feasibility: Cost estimation and ROI analysis.   + Legal feasibility: Compliance with regulations (e.g., GDPR).   + Operational feasibility: Alignment with organizational goals.   + Schedule feasibility: Timelines for delivery.   **2. Impact of Feasibility Studies on Project Success**  **Assessment Criteria:**   * Assess how different feasibility criteria influence decision-making during software investigations. * Provide examples from your chosen scenario to illustrate their impact on:   + Project planning.   + Resource allocation.   + Risk mitigation.   **Specific Identified Feasibility Criteria:**   * Evaluate how legal, social, economic, technical, and organizational constraints affect project outcomes. |
| **Software Development Plan (SDP) template** |
| This **Software Development Plan (SDP) template** is structured to reflect real-world practices, ensuring alignment with professional standards while addressing the learning outcomes (LO1 and LO2) and assessment criteria outlined in the assignment brief. It includes all essential components commonly found in practical SDP documents and additional sections tailored to meet academic requirements**.**  **1. Project Overview**  **1.1 Purpose**   * Define the primary purpose of the software project and its expected outcomes. * Highlight the business need or problem being addressed.   **1.2 Scope**   * Clearly outline the scope of the project, including:   + Features and functionalities to be delivered.   + Boundaries of the project (what is included and excluded).   + Target audience and stakeholders.   **1.3 Objectives**   * Specify measurable objectives for the project, such as:   + Improving efficiency in booking systems.   + Reducing delivery times for food orders.   + Enhancing student engagement in LMS platforms.   + Streamlining patient record management.   **2. Software Development Lifecycle Models**  **2.1 Overview of Lifecycle Models**   * Describe two iterative lifecycle models (e.g., Agile, Spiral) and two sequential models (e.g., Waterfall, V-Model). * Provide diagrams illustrating workflows, stages, and processes for each model. * Discuss their suitability for specific project environments.   **2.2 Risk Management in SDLC Models**   * Identify risks associated with each lifecycle model:   + Technical risks (e.g., system compatibility issues).   + Resource risks (e.g., budget constraints).   + Process risks (e.g., scope creep in iterative models). * Propose strategies for managing these risks:   + Risk identification techniques.   + Mitigation plans.   + Monitoring processes.   **2.3 Recommendation of an SDLC Model**   * Recommend an appropriate lifecycle model for your chosen project scenario. * Justify your choice by discussing:   + Benefits and drawbacks of the selected model.   + Appropriateness based on project requirements, constraints, and expected outcomes.   **2.4 Assessment of Waterfall Model for Large Projects**   * Evaluate the merits of applying the Waterfall lifecycle model to large-scale projects. * Compare its suitability against other models in terms of:   + Stability and predictability.   + Resource management.   + Scalability for complex projects.   **3. Feasibility Study**  **3.1 Purpose of a Feasibility Study**   * Explain why feasibility studies are essential for software development projects. * Highlight their role in reducing risks, ensuring project viability, and supporting decision-making processes.   **3.2 Comparison of Technical Solutions**   * Describe methods for comparing technical solutions, such as:   + Cost-benefit analysis.   + Decision matrices. * Provide examples relevant to your chosen project scenario.   **3.3 Components of a Feasibility Report**   * Discuss key components such as:   + Technical feasibility: Compatibility with existing systems or technologies.   + Economic feasibility: Cost estimation and ROI analysis.   + Legal feasibility: Compliance with regulations (e.g., GDPR).   + Operational feasibility: Alignment with organizational goals and workflows.   + Schedule feasibility: Timelines for delivery.   **3.4 Impact of Feasibility Criteria on Project Success**   * Assess how different feasibility criteria influence decision-making during software investigations. * Provide examples from your chosen scenario to illustrate their impact on:   + Project planning.   + Resource allocation.   + Risk mitigation.   **4. Functional Requirements**  **4.1 Front-End Functionalities**  Define user-facing features such as:   1. Responsive design compatible with desktops, tablets, and mobile devices. 2. Search functionalities tailored to specific needs:    * For hotels: Room availability by date or location.    * For food delivery: Menu items by category or price range.    * For LMS: Courses by department or instructor name.   **4.2 Back-End Functionalities**  Define system-level functionalities such as:   1. Data management capabilities for administrators to add, edit, or delete records (e.g., bookings, orders, patient details). 2. Secure authentication systems with role-based access control:    * Customers can only access their profiles.    * Administrators can manage all data within their scope.   **5. Non-Functional Requirements**  **5.1 Performance**  Example: Ensure fast response times (<2 seconds per transaction) under normal traffic conditions.  **5.2 Scalability**  Example: Design architecture to support future growth in terms of user base and additional features.  **5.3 Security**  Example:  Encrypt sensitive data using industry-standard protocols (e.g., AES encryption). Protect against vulnerabilities such as SQL injection or cross-site scripting (XSS).  **5.4 Reliability**  Ensure system uptime exceeds 99% monthly with robust error-handling mechanisms.  **6. System Design**  **6.1 Diagrams**  Include diagrams such as:   1. Entity Relationship Diagrams (ERDs): Represent database structure clearly and effectively. 2. Data Flow Diagrams (DFDs): Illustrate how data flows between different modules within the system.   **7. Risk Management**  **7.1 Identification of Risks**  Identify potential risks specific to your chosen project scenario:   1. Security vulnerabilities in data storage or transmission. 2. Delays in project timelines due to resource constraints.   **7.2 Mitigation Strategies**  Propose strategies to address these risks effectively:   1. Implementing robust security protocols like SSL encryption. 2. Allocating contingency resources for unexpected delays.   **8. Timeline & Milestones**  Define key milestones for the project lifecycle:   1. Requirement gathering completion date. 2. Feasibility study submission date. 3. Design phase completion date. 4. Testing phase start and end dates.   Provide a Gantt chart or similar visualization to track progress against milestones.  **9. Quality Assurance Plan**  Outline testing strategies such as:   1. Unit testing for individual components. 2. Integration testing across modules. 3. User acceptance testing before deployment.   Specify tools or frameworks used for testing (e.g., Selenium).  **10. Communication Plan**  Define communication protocols among stakeholders:   1. Weekly progress reports via email or meetings. 2. Use of collaboration tools like Jira or Trello for task tracking.   **11. Conclusion**  Summarize key findings from lifecycle analysis and feasibility study sections: Highlight insights gained from evaluating SDLC models and feasibility criteria while providing recommendations for implementing your selected lifecycle model effectively.  **12. References**  Include all sources cited using Harvard referencing style: - Textbooks - Journals - Websites  **13. Appendices**  - Attach supporting materials such as: - Diagrams - Tables - Additional project-related data |

|  |  |  |
| --- | --- | --- |
| **­­** | | |
| **Pass** | **Merit** | **Distinction** |
| **LO1** Describe different software development lifecycles | | **D1** Assess the merits of applying the Waterfall lifecycle model to a large software development project. |
| **P1** Describe two iterative and two sequential software lifecycle models.  **P2** Explain how risk is managed in software lifecycle models. | **M1** Discuss using an example, why a particular lifecycle model is selected for a development environment. |
| **LO2** Explain the importance of a feasibility study | | **D2** Assess the impact of different feasibility criteria on a software investigation. |
| **P3** Explain the purpose of a feasibility report.  **P4** Describe how technical solutions can be compared. | **M2** Discuss the components of a feasibility report. |