**ВСТУП**

Проектування програмного забезпечення для електронного магазину - це складний процес, що вимагає кілька етапів. Нижче наведено загальний огляд кожного етапу проектування, а також вказівки щодо важливих кроків на кожному з них:

Відкриття:

Намалюйте карту подорожі клієнта до майбутнього рішення та процесу: Розробіть діаграму, що показує всі етапи від вибору продукту до оформлення замовлення і доставки. Визначте ключові кроки та взаємозв'язки між ними.

Узагальнення основних вимог: Проведіть аналіз потреб бізнесу, користувачів та функціональних і нефункціональних вимог. Це включає в себе визначення, які функції має виконувати система, і які параметри її продуктивності та безпеки.

Визначення ключових ризиків та нормативних вимог: Визначте потенційні ризики, які можуть виникнути під час розробки та експлуатації електронного магазину. Дотримання нормативних вимог, таких як законодавство про захист даних користувачів, також важливо.

Визначте архітектурно значущі вимоги: Виокреміть ті вимоги, які суттєво впливають на архітектуру системи, такі як масштабуваність, безпека, швидкодія, інтеграція з іншими системами тощо.

Розробіть архітектуру: Створіть високорівневий огляд архітектури вашої системи. Включіть в нього інформацію про ІТ-середовище, вибір платформи, основні патерни проектування та інтеграції з іншими системами. Розгляньте альтернативи і визначте їх переваги та недоліки.

Перелічіть експерименти: Проведіть дослідження або експерименти, щоб краще зрозуміти вимоги та вибрати оптимальний шлях. Наприклад, можна провести опитування користувачів для визначення їхніх потреб і побажань.

**Опис методів та інструментів**

Під час проектування електронного магазину важливо використовувати різні методи та інструменти для розробки та аналізу системи. Ось короткий опис деяких основних методів та інструментів, які можуть бути використані:

Візуальне моделювання:

UML (Unified Modeling Language): UML - це стандартна мова для візуального моделювання систем. Вона дозволяє створювати діаграми, такі як діаграми варіантів використання, діаграми класів, діаграми послідовності, що допомагають визначити архітектуру та взаємодію компонентів системи.

Мови специфікацій:

HTML/CSS: HTML і CSS використовуються для розробки веб-інтерфейсу електронного магазину. HTML визначає структуру сторінок, а CSS - їх зовнішній вигляд.

Діаграми потоків:

Діаграми потоків даних (DFD): DFD використовуються для моделювання потоків даних в системі та їх обробки. Вони допомагають зрозуміти, як дані пересуваються в системі та які процеси їх обробляють.

Методології розробки:

Agile: Методологія Agile дозволяє розробникам працювати над проектом у невеликих ітераціях, дозволяючи швидко внести зміни та пристосовуватися до зміни вимог клієнта.

Waterfall: Методологія Waterfall передбачає послідовний розробку проекту, де кожен етап завершується перед початком наступного. Ця методологія підходить для проектів зі статичними вимогами.

Інструменти управління проектом:

JIRA: JIRA - це інструмент для управління проектами та відстеження задач. Він допомагає зберігати і відстежувати завдання та звіти про проект.

Прототипування:

MockFlow, Balsamiq: Інструменти для створення прототипів інтерфейсу, які дозволяють створити ескизи сторінок та взаємодію клієнта з системою.

Мови програмування:

JavaScript, PHP, Python: Використовуються для розробки функціональності магазину та взаємодії з базою даних.

**Специфікація проекту**

Специфікація проекту є важливим документом, який допомагає детально визначити всі вимоги та основні аспекти розробки програмного забезпечення для електронного магазину. Нижче представлений загальний огляд того, що повинно включати специфікацію проекту:

Встановлення вимог:

Опис пари Бізнес-вимог: Це загальний опис цілей і вимог, які ставляться перед проектом електронного магазину.

Опис пари Вимог користувача для Бізнес-вимоги: Визначення конкретних потреб користувачів, які випливають з бізнес-вимог.

Опис пари Функціональних вимог рішення для Вимог користувача: Конкретні функції та можливості, які повинні бути реалізовані в системі для задоволення вимог користувачів.

Опис пари Нефункціональних вимог рішення для Вимог користувача: Нефункціональні вимоги, такі як швидкодія, безпека, доступність, які також повинні бути враховані.

Діаграма варіантів використання: Графічне представлення сценаріїв взаємодії користувача з системою.

Словесні специфікації варіантів використання: Детальні описи кожного варіанту використання з точки зору користувача.

Діаграма класів: Визначення структури об'єктів системи та їх взаємозв'язків.

Моделювання поведінки системи:

Діаграма діяльності: Показує послідовність дій в конкретному сценарії.

Діаграма послідовності: Візуально представляє взаємодію між різними об'єктами системи в різних сценаріях.

Проектування архітектури програмного забезпечення:

Діаграма розгортання: Показує, як компоненти системи розташовані на апаратному рівні.

Діаграма компонентів: Представляє структуру програмного забезпечення та взаємозв'язки між його компонентами.

Малюнок архітектури: Візуальне представлення загальної архітектури системи.

Проектування бази даних:

ER-модель: Показує сутності, їх атрибути та взаємозв'язки в базі даних.

Логічний дизайн: Визначення структури таблиць та зв'язків в базі даних.

Фізичний дизайн: Вибір конкретної СУБД, оптимізація запитів та інші аспекти фізичної реалізації бази даних.

Інформаційні запити: Опис основних запитів до бази даних, які повинні бути підтримані.

Генерація коду або створення прототипу - це наступний крок після завершення проектування і створення специфікації проекту. Нижче розглянуті два підходи до цього етапу:

Генерація коду:

Ручна розробка: Розробники можуть розпочати ручну розробку програмного коду на основі специфікації проекту. Цей метод надає повний контроль над кодом, але може зайняти багато часу та зусиль.

Використання кодогенераторів: Кодогенератори - це інструменти, які автоматично створюють програмний код на основі визначених моделей або шаблонів. Наприклад, для розробки веб-додатків ви можете використовувати фреймворки, які генерують код на основі специфікацій.

Використання IDE (середовища розробки): Деякі інтегровані середовища розробки надають можливість автоматично генерувати код для конкретних частин програми на основі специфікацій.

Створення прототипу:

Прототип - це спрощена версія програми, яка дозволяє швидко оцінити функціональність та взаємодію з користувачем без докладного розробки всіх компонентів. Він дозволяє отримати зворотний зв'язок від користувачів і зацікавлених сторін щодо дизайну та функціональних можливостей.

Прототипування може бути виконане за допомогою спеціалізованих інструментів для створення інтерфейсу користувача, таких як Axure RP, Sketch, Figma, або навіть за допомогою HTML/CSS для створення інтерактивних веб-прототипів.