**Программное обеспечение для автоматизации процесса разработки. CASE-технологии**

**Корпоративная информационная система (КИС)**

**1. ТРЕБОВАНИЯ К CASE-СРЕДСТВАМ**

**1.1. Функциональные требования к CASE-технологии:**

* Поддержка полного жизненного цикла разработки
* Возможность коллективной работы над проектом
* Интеграция с системами управления требованиями
* Поддержка визуального моделирования
* Автоматическая генерация кода
* Реверс-инжиниринг существующих систем

**1.2. Критерии выбора CASE-средств:**

* Поддержка нотаций UML 2.x
* Интеграция со средами разработки Java и .NET
* Совместимость с реляционными СУБД
* Возможность документирования моделей
* Поддержка командной разработки

**2. СТРУКТУРА И СОДЕРЖАНИЕ РЕПОЗИТОРИЯ**

**2.1. Структура репозитория:**

КИС\_Проект/

├── Модели/

│ ├── Бизнес-модели/

│ ├── Модели данных/

│ └── Модели компонентов/

├── Требования/

├── Документация/

└── Исходный\_код/

**2.2. Физическая реализация:**

* **СУБД:** PostgreSQL 14+
* **Сервер приложений:** Apache Tomcat 9+
* **Клиентские приложения:** Eclipse Modeling Tools

**2.3. Методы доступа:**

* Веб-интерфейс для просмотра моделей
* API для интеграции с IDE
* Клиентские приложения для моделирования

**3. ГРАФИЧЕСКИЙ ЯЗЫК МОДЕЛИРОВАНИЯ**

**3.1. Используемые нотации:**

* UML 2.5 (Unified Modeling Language)
* BPMN 2.0 (Business Process Model and Notation)
* ERD (Entity-Relationship Diagrams)

**3.2. Виды диаграмм и их назначение:**

**Диаграммы структуры:**

* **Диаграмма классов:** проектирование объектной модели
* **Диаграмма компонентов:** архитектура системы
* **Диаграмма развертывания:** инфраструктура

**Диаграммы поведения:**

* **Диаграмма вариантов использования:** функциональные требования
* **Диаграмма последовательностей:** взаимодействие объектов
* **Диаграмма деятельности:** бизнес-процессы

**4. ОРГАНИЗАЦИЯ КОЛЛЕКТИВНОЙ РАЗРАБОТКИ**

**4.1. Подходы к организации:**

* Ролевая модель разработки
* Распределение ответственности
* Регулярные код-ревью

**4.2. Виды взаимодействия:**

* Ежедневные стендапы
* Еженедельные планирования
* Система управления задачами (Jira)
* Система контроля версий (Git)

**5. ВОЗМОЖНОСТИ CASE-СРЕДСТВ**

**5.1. Автоматическая генерация кода:**

* Генерация классов на Java/C# из UML-моделей
* Создание SQL-скриптов из ER-диаграмм
* Генерация документации из моделей

**5.2. Макетирование и прототипирование:**

* Создание прототипов интерфейсов
* Генерация HTML-форм из моделей
* Быстрое прототипирование бизнес-процессов

**6. ИНТЕГРИРОВАННЫЕ СРЕДЫ РАЗРАБОТКИ**

**6.1. Используемые ИСР:**

* **Enterprise Architect:** комплексное моделирование
* **Visual Paradigm:** проектирование БД и интерфейсов
* **IntelliJ IDEA:** разработка бизнес-логики

**6.2. Способы применения:**

* Реверс-инжиниринг существующего кода
* Синхронизация моделей и кода
* Автоматизация тестирования

**7. ДОКУМЕНТ ПО ПРИМЕНЕНИЮ CASE-ТЕХНОЛОГИИ**

**7.1. Стадии жизненного цикла и применяемые средства:**

**Стадия анализа:**

* **Средства:** Enterprise Architect, BPMS
* **Результаты:** модели бизнес-процессов, требования

**Стадия проектирования:**

* **Средства:** Visual Paradigm, ERwin Data Modeler
* **Результаты:** архитектурные модели, схемы БД

**Стадия реализации:**

* **Средства:** IntelliJ IDEA, генераторы кода
* **Результаты:** исходный код, компоненты системы

**Стадия тестирования:**

* **Средства:** JUnit, Selenium
* **Результаты:** тестовые сценарии, отчеты

**8. АНАЛИЗ ТРЕБОВАНИЙ К КЛИЕНТСКОМУ ПРИЛОЖЕНИЮ**

**8.1. Функциональные требования:**

* Авторизация и аутентификация пользователей
* Управление производственными заказами
* Просмотр складских остатков
* Формирование отчетов
* Управление профилем пользователя

**8.2. Требования к пользовательскому интерфейсу:**

* Адаптивный веб-дизайн
* Поддержка мобильных устройств
* Интуитивная навигация
* Быстрый отклик на действия пользователя

**9. ПРОЕКТ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА**

**9.1. Макеты экранных форм:**

* **Главная страница:** дашборд с ключевыми показателями
* **Модуль заказов:** список, создание, редактирование заказов
* **Складской модуль:** учет остатков, движения товаров
* **Отчеты:** аналитические панели, графики, экспорт

**9.2. Элементы интерфейса:**

* Верхнее меню навигации
* Боковая панель быстрого доступа
* Область основного контента
* Статусная строка

**10. РАЗРАБОТКА ПРОТОТИПА КЛИЕНТСКОГО ПРИЛОЖЕНИЯ**

**10.1. Технологический стек:**

* **Frontend:** React.js, TypeScript, Material-UI
* **Backend:** Spring Boot, Java 17
* **База данных:** PostgreSQL 14
* **Сборка:** Maven, Webpack

**10.2. Реализация функционала:**

* Компоненты React для интерфейса
* REST API для взаимодействия с сервером
* Сервисы бизнес-логики на Spring
* JPA/Hibernate для доступа к данным

**11. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ**

**11.1. Виды тестирования:**

* **Модульное тестирование:** JUnit, Mockito
* **Интеграционное тестирование:** TestContainers
* **Системное тестирование:** Selenium, Cypress
* **Нагрузочное тестирование:** JMeter

**11.2. Контрольные тесты:**

* Тестирование авторизации
* Проверка создания заказа
* Тестирование отчетов
* Проверка валидации данных

**12. ВЕРИФИКАЦИЯ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ**

**12.1. Сравнение с требованиями:**

* Проверка соответствия функциональным требованиям
* Тестирование нефункциональных характеристик
* Валидация пользовательского опыта

**12.2. Метрики качества:**  
-тестов: >80%

* Время отклика: < 2 секунды
* Доступность системы: 99.5%
* Удовлетворенность пользователей: >4.5/5

**ВЫВОДЫ**

1. Разработана комплексная методология применения CASE-технологий для автоматизации процесса разработки КИС
2. Определены требования к инструментальным средствам и репозиторию проекта
3. Спроектирована архитектура клиентского приложения с использованием современных технологий
4. Реализован прототип системы с полным циклом тестирования
5. Проведена верификация функциональных возможностей разработанного приложения

**Результат:** Приобретены практические навыки применения CASE-технологий и современных средств автоматизации для разработки сложных корпоративных информационных систем.