Курс: Otus Software Architect
Домашнее задание 2

по кате Нила Форда “Lights, Please”

Выполнил: Крошкин Игорь

23.01.2024

Содержание

1. Бизнес-контекст

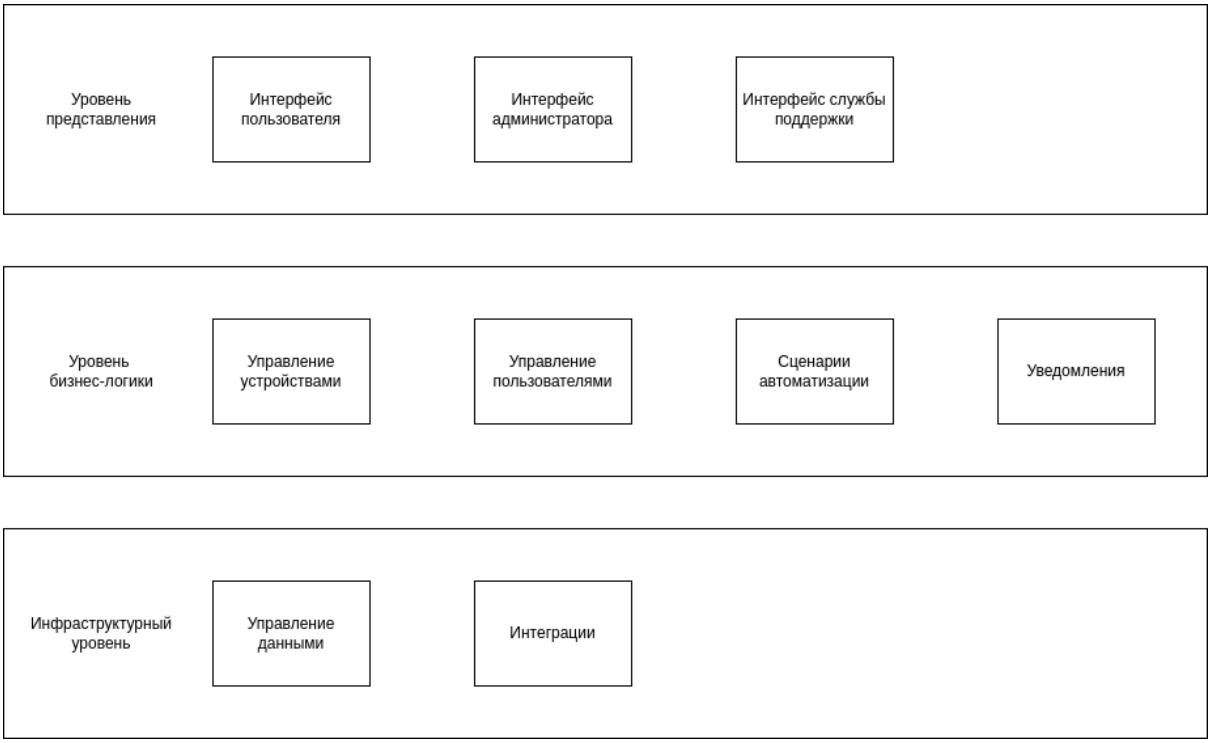
Оригинал: https://nealford.com/katas/kata?id=LightsPlease

Гигант в сфере бытовой электроники хочет создать систему для автоматизации дома: включение и выключение света, запирание и отпирание дверей, удаленное наблюдение с помощью камер и неопределенное поведение в будущем.

- Пользователи: каждая система будет продаваться потребителям (небольшим семьям), но компания рассчитывает продать тысячи таких устройств в течение первых трех лет.
- Требования:
    - система должна быть максимально готова к эксплуатации, но при этом продаваться в модульных блоках (камера, замок, термостат и т. д.) для удобства покупки
    - устройства должны быть доступны через Интернет (для удаленного мониторинга и доступа), и предполагается, что у пользователя будет существующая настройка WiFi (маршрутизатор и подключение) для подключения
    - клиенты могут программировать систему для управления различными модулями в соответствии со своими потребностями.
    - электротехникой для блоков займутся другие группы, а программные протоколы для управления модулями будут гибкими в соответствии с потребностями/проектами вашей архитектуры. (Они займутся реализацией модульной части протокола, как только вы им это укажете.)
- Дополнительный контекст:
    - готов инвестировать большую сумму, чтобы запустить это новое направление бизнеса
    - собирает данные от клиентов, которые согласились собирать более широкую статистику
    - международная компания

2.1. Декомпозиция по функциональным модулям

Рис.2.1. Декомпозиция по функциональным модулям в слоистой архитектуре.



При данном способе декомпозиции система разделяется на модули, каждый из которых отвечает за определённую функциональность (например, управление устройствами, автоматизация, уведомления).

1. Интерфейс пользователя - обеспечивает взаимодействие пользователя с системой через приложения:

- Мобильное приложение (iOS, Android) и веб-интерфейс
- Визуализация данных (состояние устройств, видеопотоки, лог событий)
- Управление устройствами и настройка сценариев
- Голосовое управление через ассистенты

2. Интерфейс администратора - обеспечивает расширенные функции для управления безопасностью и интеграциями, которые доступны всем пользователям системы:

- Расширенное управление пользователями, устройствами
- Настройка шаблонов пользовательского интерфейса
- Проведение диагностики системы, включая состояние устройств, пользователей и сценариев
- Операции, связанные с обновлениями системы.

3. Интерфейс службы поддержки - обеспечивает взаимодействие сотрудника службы поддержки с пользователем:

- Веб-интерфейс
- Просмотр поступивших запросов
- Фильтрация запросов по приоритету
- Управление запросами на основе обращений
- Отслеживание истории запросов от каждого пользователя

4. Управление устройствами - обеспечивает взаимодействие с физическими устройствами умного дома:

- Управление освещением (включение/выключение, регулировка яркости, изменение цвета)
- Контроль замков (запирание/отпирание дверей, мониторинг состояния)
- Управление камерами (включение/выключение, запись, поворотные механизмы, доступ к видеопотоку)
- Поддержка стандартов и протоколов (Zigbee, Z-Wave, Wi-Fi, Bluetooth, MQTT)
- Подключение новых устройств (прошивка, настройка, удаление)

5. Управление пользователями - контролирует доступ пользователей к системе:

- Аутентификация (пароль, биометрия, двухфакторная аутентификация)
- Управление ролями (администратор, пользователь, гость)
- Настройка разрешений на управление отдельными устройствами
- Логирование и мониторинг действий пользователей

6. Сценарии автоматизации - реализует автоматизацию и сценарии для умного дома:

- Создание пользовательских сценариев (например, «включить свет при движении в комнате»)
- Реакция на события (например, срабатывание датчика движения или открытие двери)
- Работа с расписанием (включение/выключение по времени)
- Настройка триггеров на основе условий (например, «закрыть двери, если дома никого нет»)

7. Уведомления - уведомляет пользователей о событиях в системе:

- Push-уведомления, SMS, email (например, о незакрытых дверях)
- Настройка приоритетов уведомлений (обычные, критические)
- Интеграция с системами оповещения (например, тревожные сигналы)

8. Управление данными - сохраняет и обрабатывает данные, поступающие от устройств и пользователей:

- Хранение истории событий (видеозаписи, действия пользователей, состояния устройств)
- Аналитика (например, энергоэффективность, популярность сценариев)
- Поддержка облачного и локального хранения данных
- Подготовка данных для искусственного интеллекта (например, предсказание поведения).

9. Интеграции - обеспечивает взаимодействие с другими платформами и системами:

- Подключение голосовых ассистентов
- Интеграция с погодными сервисами (для автоматизации на основе прогноза)
- Взаимодействие с системами безопасности (например, вызов службы охраны)
- Поддержка API для интеграции сторонних устройств

2.2. Оценка изменений

В рамках декомпозиции системы по функциональным модулям можно определить наиболее вероятные сценарии изменений, оценить их вероятность и стоимость реализации, см.Таблица 2.2.

Таблица 2.2. Оценка изменений при декомпозиции по функциональным модулям.

Сценарий изменения	Модуль	Вероятность изменения	Стоимость изменения	Комментарий
Добавление нового типа устройства	Управление устройствами	Высокая	Средняя	Потребуется адаптация API для поддержки новых протоколов
Обновление пользовательского интерфейса	Интерфейс пользователя	Высокая	Средняя	Изменения в визуальном дизайне и настройке пользовательских сценариев
Интеграция с новым голосовым ассистентом	Интеграции	Средняя	Высокая	Требует разработки новых API

Добавление сценариев автоматизации	Сценарии автоматизации	Высокая	Средняя	Требуется добавление новых условий и действий
Масштабирование системы для многоквартирных домов	Все модули	Низкая	Очень высокая	Потребуется переработка архитектуры для работы с несколькими изолированными подсистемами
Поддержка новых каналов уведомлений	Уведомления	Средняя	Низкая	Потребует минимальных изменений
Поддержка дополнительных языков интерфейса	Интерфейс пользователя	Низкая	Низкая	Добавление локализации через языковые файлы
Реализация рекомендаций на основе ИИ	Сценарии автоматизации	Низкая	Очень высокая	Интеграция алгоритмов машинного обучения и переработка логики сценариев

3.1. Декомпозиция по пользовательским сценариям

Рис.3.1. Декомпозиция по пользовательским сценариям.

При данном способе декомпозиции система разделяется на сценарии взаимодействия пользователя с системой (например, управление освещением, режимами присутствия и безопасности).

1. Управление освещением:

- Включение/выключение света через мобильное приложение
- Автоматическое включение света при обнаружении движения в комнате
- Регулировка яркости и цвета освещения (например, теплый свет вечером, холодный утром)
- Сценарии освещения для определенных случаев (например, «Кино», «Вечеринка», «Работа»)
- Автоматическое выключение света в помещении при отсутствии движения в течение заданного времени

2. Управление дверными замками:

- Запирание/отпирание дверей с помощью мобильного приложения
- Автоматическое запирание дверей при выходе из дома (на основе геолокации или сценария)

- Предоставление временного доступа (например, для курьеров или гостей)
- Уведомление при попытке взлома или открытии двери без авторизации
- История действий (кто и когда открыл или закрыл дверь)

3. Управление климатом:

- Автоматическое регулирование температуры (например, снижение при отсутствии владельцев)
- Настройка температуры для разных помещений
- Уведомления о превышении заданного диапазона температуры или влажности
- Интеграция с системами кондиционирования и отопления

4. Управление камерами:

- Просмотр видеопотока с камер через мобильное приложение в реальном времени
- Уведомление о движении в поле зрения камеры (например, в отсутствие владельцев)
- Запись видео при обнаружении движения и сохранение в облаке или локально
- Настройка зон наблюдения (например, игнорирование движения домашних животных)
- Включение камер при активировании системы безопасности

5. Сценарии автоматизации:

- Создание сценариев:
    - «Доброе утро»: Включение света, открытие штор, включение кофеварки
    - «Выход из дома»: Выключение света, запирание дверей, активация охранного режима
    - «Добро пожаловать»: Включение света в прихожей и снятие с охраны при входе
- Автоматизация на основе времени (например, выключение света в 23:00).
- Автоматизация на основе внешних факторов (например, закрытие окон при дожде).
- Подключение погодных данных для сценариев (например, включение обогрева при падении температуры)

6. Уведомления

- Уведомление о незакрытых дверях или окнах
- Уведомление о срабатывании датчиков (например, движения, дыма, протечки воды)

- Напоминания о регулярных действиях (например, замена батареи в датчиках)
- Уведомление о завершении сценария (например, «Все устройства выключены»)

7. Управление через голосовых ассистентов
- Включение/выключение света голосовой командой
- Открытие или закрытие дверей по голосовой команде
- Запуск сценариев (например, «Активация охраны»)
- Запрос информации (например, «Какая температура в доме?»)

8. Режимы присутствия
- Имитация присутствия (включение света или проигрывание звуков при отсутствии владельцев)
- Смена режимов (например, «Дома», «Нет дома», «Ночь»)
- Настройка сценариев на основе режима (например, все устройства отключаются при переходе в режим «Нет дома»)

9. Режимы безопасности
- Постановка дома на охрану (все двери заперты, камеры активны)
- Уведомление о попытке несанкционированного доступа
- Уведомление служб охраны или полиции при обнаружении вторжения
- Просмотр истории событий, связанных с безопасностью (например, срабатывание датчиков)

3.2. Оценка изменений

В рамках декомпозиции системы по пользовательским сценариям можно определить наиболее вероятные сценарии изменений, оценить их вероятность и стоимость реализации:

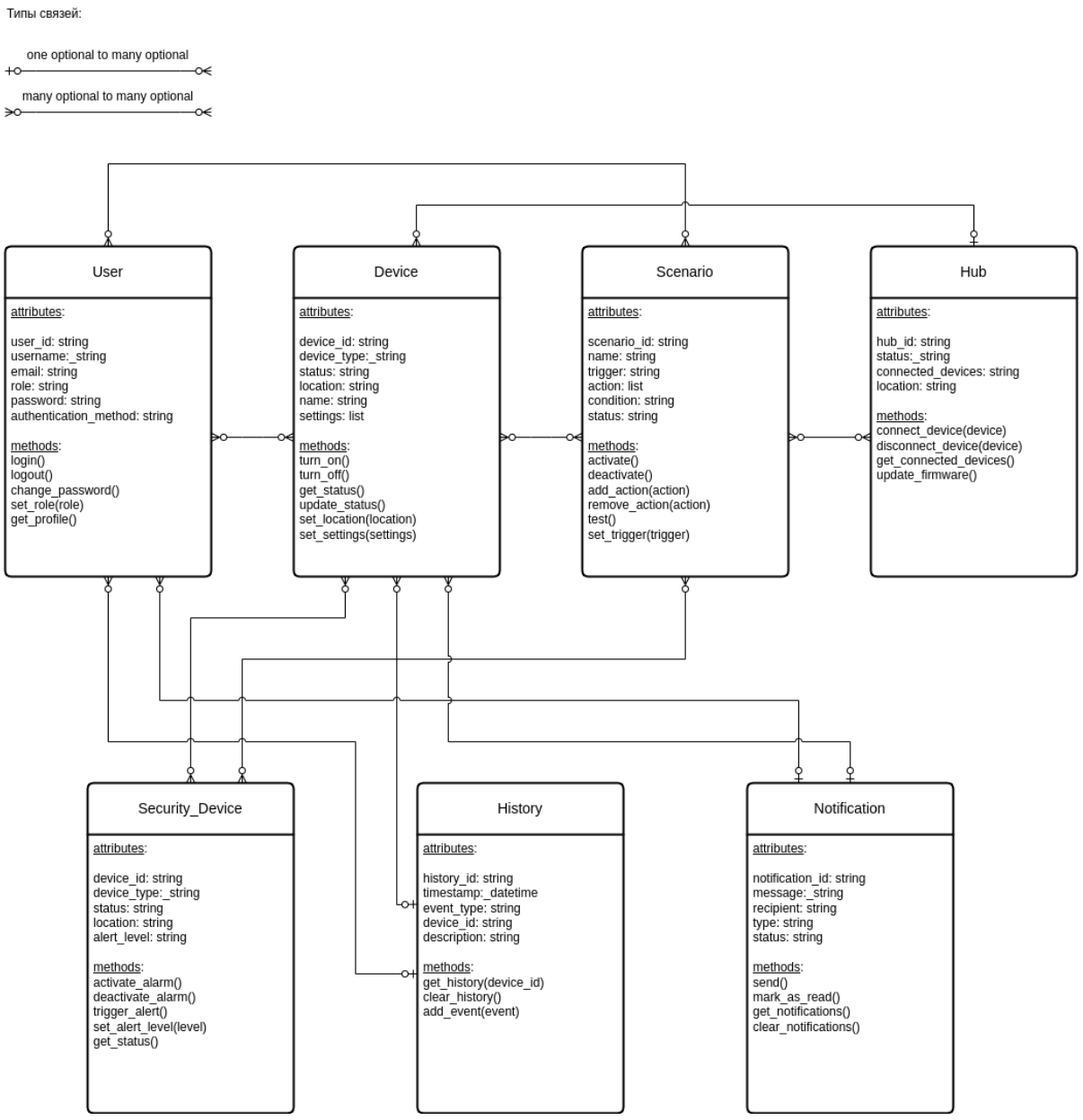
Таблица 3.2.1. Оценка изменений при декомпозиции по пользовательским сценариям.

Сценарий изменения	Пользовательский сценарий	Вероятность изменения	Стоимость изменения	Комментарий
Добавление нового типа устройства	Сценарии автоматизации	Высокая	Средняя	Потребуется адаптация API и обновление сценариев автоматизации
Обновление	Сценарии	Высокая	Средняя	Изменения

пользовательского интерфейса	автоматизации			могут коснутся нескольких пользовательских сценариев
Интеграция с новым голосовым ассистентом	Управление через голосовых ассистентов	Средняя	Средняя	Поддержка API стороннего ассистента
Добавление сценариев автоматизации	Сценарии автоматизации	Высокая	Низкая	Добавление шаблонов не требует сложной доработки
Масштабирование системы для многоквартирных домов	Все	Низкая	Очень высокая	Требует переработки архитектуры для поддержки изолированных подсистем
Поддержка новых каналов уведомлений	Уведомления	Средняя	Низкая	Добавление интеграции с новыми каналами
Поддержка дополнительных языков интерфейса	Все	Низкая	Низкая	Локализация интерфейса через языковые файлы
Реализация рекомендаций на основе ИИ	Все	Средняя	Очень высокая	Интеграция сбора данных о поведении пользователей и алгоритмов машинного обучения

4. Модель предметной области

Рис. 4.1. Модель предметной области системы управления умным домом “Lights, Please”



В результате декомпозиции предметной области можно выделить следующие ключевые сущности с описанием атрибутов, методов и связи между ними:

1. Пользователь (User)

Атрибуты:

- user_id (строка): Уникальный идентификатор пользователя
- username (строка): Имя пользователя
- email (строка): Адрес электронной почты пользователя
- role (строка): Роль пользователя (администратор, пользователь, гость)
- password (строка): Пароль пользователя
- authentication_method (строка): Метод аутентификации (пароль, биометрия и т.д.)

Методы:

- login(): Авторизация пользователя
- logout(): Выход пользователя
- change_password(new_password): Изменить пароль пользователя
- set_role(role): Установить роль пользователя (например, администратор, гость)
- get_profile(): Получить информацию о профиле пользователя

2. Устройство (Device)

Атрибуты:

- device_id (строка): Уникальный идентификатор устройства
- device_type (строка): Тип устройства (освещение, замок, камера и т.д.)
- status (строка): Текущее состояние устройства (включено/выключено, заблокировано/открыто)
- location (строка): Местоположение устройства (например, «Кухня», «Гостиная»)
- name (строка): Название устройства, заданное пользователем (например, «Лампа в спальне»)
- settings (объект): Дополнительные параметры устройства (яркость, цвет для освещения, температура для термостата)

Методы:

- turn_on(): Включить устройство
- turn_off(): Выключить устройство
- update_status(): Обновить статус устройства
- set_location(location): Установить местоположение устройства
- set_settings(settings): Настроить параметры устройства (яркость, температура и т.д.)
- get_status(): Получить текущее состояние устройства

3. Сценарий (Scenario)

Атрибуты:

- scenario_id (строка): Уникальный идентификатор сценария
- name (строка): Название сценария (например, «Доброе утро», «Выход из дома»)

- trigger (строка): Условие или триггер (например, «время», «датчик движения»)
- action (список объектов): Список действий, которые должны быть выполнены (например, включение/выключение устройств)
- condition (строка): Условие, при котором сценарий должен быть активирован (например, «если дома никого нет»)
- status (строка): Состояние сценария (активен/неактивен)

Методы:

- activate(): Активировать сценарий (например, в случае срабатывания триггера)
- deactivate(): Деактивировать сценарий
- add_action(action): Добавить действие в сценарий
- remove_action(action): Удалить действие из сценария
- test(): Протестировать сценарий (проверить, правильно ли выполняются действия)
- set_trigger(trigger): Установить триггер для активации сценария

4. Устройство безопасности (Security_Device)

Атрибуты:

- device_id (строка): Уникальный идентификатор устройства безопасности.
- device_type (строка): Тип устройства безопасности (например, датчик движения, камера, сигнализация).
- status (строка): Состояние устройства (активно/неактивно).
- location (строка): Местоположение устройства (например, «Коридор», «Двор»)
- alert_level (строка): Уровень тревоги (низкий, средний, высокий)

Методы:

- activate_alarm(): Активировать сигнал тревоги
- deactivate_alarm(): Деактивировать сигнал тревоги
- trigger_alert(): Сгенерировать уведомление о тревоге
- set_alert_level(level): Установить уровень тревоги для устройства
- get_status(): Получить состояние устройства безопасности

5. История (History)

Атрибуты:

- history_id (строка): Уникальный идентификатор записи
- timestamp (дата/время): Время, когда событие произошло
- event_type (строка): Тип события (включение устройства, срабатывание датчика, изменение состояния системы)
- device_id (строка): Идентификатор устройства, которое вызвало событие
- description (строка): Описание события

Методы:

- get_history(device_id): Получить историю событий для определенного устройства
- clear_history(): Очистить историю событий
- add_event(event): Добавить новое событие в историю

6. Уведомление (Notification)

Атрибуты:

- notification_id (строка): Уникальный идентификатор уведомления
- message (строка): Сообщение уведомления
- recipient (строка): Получатель уведомления (например, пользователь, группа пользователей)
- type (строка): Тип уведомления (например, тревога, системное сообщение, напоминание)
- status (строка): Статус уведомления (непрочитанное/прочитанное)

Методы:

- send(): Отправить уведомление пользователю
- mark_as_read(): Пометить уведомление как прочитанное
- get_notifications(): Получить список всех уведомлений для пользователя
- clear_notifications(): Очистить все уведомления

7. Хаб (Hub)

Атрибуты:

- hub_id (строка): Уникальный идентификатор хаба
- status (строка): Статус хаба (активен/неактивен)
- connected_devices (список устройств): Список устройств, подключенных к хабу
- location (строка): Местоположение хаба (например, «Сервант», «Мастерская»)

Методы:

- connect_device(device): Подключить новое устройство к хабу
- disconnect_device(device): Отключить устройство от хаба
- get_connected_devices(): Получить список подключенных устройств
- update_firmware(): Обновить прошивку хаба

5. Сравнительный анализ

Каждый из способов имеет свои преимущества и недостатки, а выбор подхода зависит от целей, сложности изменений и контекста использования системы.

1. Декомпозиция по функциональным модулям

Преимущества:

- Хорошо подходит для изменений, которые касаются конкретных функций или технических аспектов (например, добавление новых типов устройств)
- Упрощает поддержку и масштабирование отдельных модулей
- Легче отслеживать влияние изменений на ограниченный набор функций

Недостатки:

- Может быть сложной для реализации изменений, затрагивающих несколько модулей (например, интеграция голосовых ассистентов)
- Требует строгой модульной архитектуры, что может увеличить время разработки

2. Декомпозиция по пользовательским сценариям

Преимущества:

- Ориентирован на улучшение пользовательского опыта
- Упрощает реализацию изменений, которые касаются работы пользователя (например, обновление интерфейса, добавление шаблонов автоматизации)
- Позволяет быстрее оценить, как изменения повлияют на конечного пользователя

Недостатки:

- Может быть сложной при реализации изменений, затрагивающих техническую часть системы
- Труднее обеспечить независимость сценариев, так как они часто пересекаются (например, управление устройствами и автоматизация)

6. Выводы

Оба способа дополняют друг друга и могут использоваться в зависимости от типа изменений.

Фокус на пользовательских сценариях повышает удобство работы, тогда как ориентация на модули упрощает техническую реализацию и поддержку системы, см.Таблица 5.1.

Таблица 5.1. Способы декомпозиции в зависимости от типа изменений.

Тип изменения	Способ декомпозиции	Причина выбора
Изменения касаются внутренней логики или технических аспектов (например, добавление API, масштабирование системы)	По функциональным модулям	Четкое разделение ответственности между модулями
Изменения направлены на улучшение пользовательского опыта	По пользовательским сценариям	Удобно оценить влияние изменений на пользователей
Изменения касаются интеграции новых технологий или устройств	По функциональным модулям	Логика интеграции сосредоточена в одном модуле
Улучшение или добавление сценариев взаимодействия пользователей с системой	По пользовательским сценариям	Ориентация на сценарии использования системы
Планируется комплексное изменение архитектуры	По функциональным модулям	Модули проще оптимизировать независимо

Для крупных систем может использоваться гибридный подход, когда ключевые изменения делятся на:

- функциональные/технические, используется модульное разбиение
- пользовательские, используется разбиение на сценарии.