**ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**им. Т.Г.ШЕВЧЕНКО**

**БЕНДЕРСКИЙ ПОЛИТЕХНИЧЕСКИЙ ФИЛИАЛ**

**Кафедра «Промышленность и информационные технологии»**

**Анализ и моделирование функциональной области. Спецификация функциональных требований**

Выполнил:

студент 3 курса, БК22АР52ИС2 группы,

дневной формы обучения, специальность:

«Информационные системы и программирование»

Герасименко Денис Андреевич

Руководитель учебной практики

Лазовский А.В.

Бендеры 2025 год.

**1. Введение**

Анализ и моделирование функциональной области — ключевые этапы в процессе разработки программного обеспечения. Эти этапы помогают понять потребности пользователей, определить функциональные требования к системе и создать модели, которые будут служить основой для дальнейшей разработки и тестирования.

**2. Анализ функциональной области**

Анализ функциональной области заключается в выявлении и детальном изучении всех аспектов, которые будут охвачены создаваемой системой. Этот этап включает в себя исследование потребностей пользователей, бизнес-процессов, а также факторов, влияющих на архитектуру и функциональность системы.

Основные задачи анализа функциональной области:

* Выявление потребностей пользователей: Определение того, что нужно пользователям и какие задачи должна решать система.
* Идентификация бизнес-процессов: Анализ существующих бизнес-процессов, которые будут поддерживаться системой.
* Определение внешних и внутренних факторов: Учет всех факторов, которые могут повлиять на систему, таких как законодательные требования, технологии и другие системы, с которыми система будет взаимодействовать.
* Построение контекстной модели: Создание модели, которая будет отражать взаимодействие системы с внешней средой и другими системами.

**2.1. Методы анализа функциональной области:**

* Интервью с пользователями: Получение информации от конечных пользователей о том, какие задачи должна решать система.
* Опросы и анкеты: Используются для сбора множественных данных о потребностях пользователей.
* Анализ существующих документов: Изучение текущих бизнес-процессов и документации, чтобы выявить требования.
* SWOT-анализ: Оценка сильных и слабых сторон системы, возможностей и угроз.
* Дельфи-метод: Использование экспертных оценок для выявления важных факторов.

**3. Моделирование функциональной области**

Моделирование функциональной области — это процесс создания моделей, которые наглядно отображают функционирование системы. Основной целью моделирования является представление системы в абстрактной форме, которая будет понятна как техническим специалистам, так и конечным пользователям.

Типы моделей, используемых при моделировании:

* Диаграммы потоков данных (DFD): Показывают, как информация движется внутри системы и какие процессы над ней выполняются. Помогают визуализировать функциональные взаимодействия.
* USE-диаграммы (диаграммы использования): Показывают, как различные пользователи или системы будут взаимодействовать с программным продуктом.
* Диаграммы классов: Используются для описания структур данных и их взаимосвязей в объектно-ориентированных системах.
* Реализация BPMN (Business Process Model and Notation): Это нотация для моделирования бизнес-процессов, отображающая взаимодействие между различными участниками процесса.

**3.1. Преимущества моделирования функциональной области:**

* Помогает понять, как система будет функционировать и как взаимодействовать с пользователями и внешними системами.
* Упрощает процесс коммуникации между различными заинтересованными сторонами, включая заказчиков, пользователей и разработчиков.
* Позволяет выявить потенциальные проблемы на ранней стадии, что способствует более качественной разработке.

**4. Спецификация функциональных требований**

Спецификация функциональных требований — это документ, который детализирует функциональные и нефункциональные требования к создаваемой системе. Этот документ является основным средством коммуникации между заказчиком, аналитиками и разработчиками.

**4.1. Основные компоненты спецификации функциональных требований:**

* Общее описание системы: Краткое описание того, что из себя представляет система и какие задачи она решает.
* Функциональные требования: Описание конкретных функций системы, которые она должна выполнять. Эти требования часто формулируются как список операций или действий, которые система должна уметь выполнять.
  + Пример: "Система должна позволять пользователю создавать, редактировать и удалять записи."
* Нефункциональные требования: Описание характеристик, которые не связаны напрямую с функциональностью, но важны для системы. Это могут быть требования к производительности, безопасности, доступности и т. д.
  + Пример: "Система должна обрабатывать не менее 1000 запросов в минуту."
* Ограничения: Условия, которые накладываются на систему. Это могут быть технические ограничения, нормативные требования, бюджетные ограничения и другие.
  + Пример: "Система должна быть совместима с операционными системами Windows и Linux."
* Интерфейсы с другими системами: Описание того, как система будет взаимодействовать с другими системами или сервисами.
  + Пример: "Система должна интегрироваться с базой данных MySQL для хранения информации о пользователях."
* Требования к безопасности: Указания на требования по защите данных, авторизации пользователей и другим аспектам безопасности.
  + Пример: "Система должна использовать двухфакторную аутентификацию при входе в систему."
* Качество и тестирование: Требования к проверке функциональности, тестированию и другим аспектам качества.
  + Пример: "Система должна пройти тестирование на 100% покрытия кода."

**4.2. Формат спецификации требований:**

* Блок-схема: Для визуализации функциональных требований и бизнес-процессов.
* Текстовое описание: Подробное описание каждого требования, как в виде обычного текста, так и с использованием таблиц и формул.
* Использование стандартов: Некоторые организации используют стандартные форматы для спецификаций, такие как IEEE 830, который помогает систематизировать процесс написания документа.

**4.3. Роль спецификации в процессе разработки:**

* Спецификация служит основой для разработки системы. На её основе создаются проектные решения, архитектура системы и технические задания.
* Документ также служит основой для тестирования, поскольку проверка функциональных требований является частью процесса верификации и валидации системы.

**5. Заключение**

Анализ и моделирование функциональной области, а также создание спецификации функциональных требований — важнейшие этапы разработки программного обеспечения. Эти процессы помогают системно подойти к созданию системы, обеспечивая точное понимание задач и требований, с которыми она должна справляться. Правильно проведённый анализ, и чётко оформленная спецификация помогают избежать множества ошибок на более поздних этапах разработки и обеспечить успешную реализацию проекта.

**4. Преимущества и недостатки использования метрик**

Преимущества:

* Управление качеством: Метрики позволяют объективно оценить качество программного продукта, помогают выявить и устранить проблемы на ранних стадиях разработки.
* Оценка рисков: Метрики помогают выявлять потенциальные риски, такие как сложности в коде или проблемы с производительностью, которые могут привести к сбоям в будущем.
* Оптимизация процессов: Использование метрик позволяет повысить эффективность разработки, оптимизировать ресурсы и время, а также улучшить координацию между командами.

Недостатки:

* Перегрузка метриками: Использование слишком большого числа метрик может привести к излишней сложности в анализе и затруднить принятие решений.
* Риск фокусирования на числах: Слишком сильное внимание к числовым показателям может привести к игнорированию других важных аспектов, таких как творческий подход или опыт команды.

**5. Заключение**

Метрики программного продукта играют важную роль в процессе разработки и поддержания ПО. Они помогают не только контролировать качество и производительность системы, но и улучшать процессы разработки, тестирования и эксплуатации. Однако важно использовать метрики сбалансированно, чтобы не перегрузить проект лишними показателями и сохранить фокус на достижении конечных целей.