**Лекция 3: Анализ и сбор требований**

**1. Введение в процесс анализа и сбора требований**

Анализ и сбор требований — один из самых критических этапов в жизненном цикле разработки информационных систем. Цель этого этапа — определить, что именно должна делать система, чтобы удовлетворить потребности бизнеса и пользователей. Ошибки, допущенные на этапе сбора и анализа требований, могут привести к значительным проблемам в дальнейшем, таким как перерасход бюджета, срывы сроков или даже провал проекта.

**Основные задачи анализа и сбора требований**:

* **Идентификация требований**: понять, какие функции должна выполнять система и какие бизнес-процессы она будет автоматизировать.
* **Документирование требований**: зафиксировать все требования к системе, включая функциональные и нефункциональные.
* **Приоритизация требований**: определить, какие требования являются наиболее важными и должны быть реализованы в первую очередь.
* **Управление требованиями**: отслеживать изменения в требованиях на протяжении всего жизненного цикла проекта.

Процесс анализа требований является ключевым для успеха проекта, так как он закладывает основу для проектирования, разработки и тестирования системы.

**2. Типы требований**

Требования можно классифицировать по разным критериям, но наиболее распространенное деление — это на функциональные и нефункциональные требования.

* **Функциональные требования** описывают конкретные функции, которые должна выполнять система. Это может быть, например, регистрация пользователей, обработка заказов, генерация отчетов, управление инвентарем и т.д. Функциональные требования описывают поведение системы и ее реакции на различные действия пользователя.

Примеры функциональных требований:

* + Система должна предоставлять пользователям возможность регистрации и авторизации.
  + Система должна поддерживать обработку платежей через кредитные карты.
  + Система должна генерировать ежемесячные финансовые отчеты.
* **Нефункциональные требования** описывают характеристики системы, такие как производительность, безопасность, удобство использования, надежность и масштабируемость. Эти требования часто называются "качественными" характеристиками системы и влияют на её общую производительность и опыт использования.

Примеры нефункциональных требований:

* + Система должна обрабатывать до 1000 запросов в минуту.
  + Время отклика системы не должно превышать 2 секунд при нагрузке до 500 пользователей одновременно.
  + Система должна поддерживать шифрование данных для обеспечения конфиденциальности.

**3. Методы сбора требований**

Для того чтобы эффективно собрать все требования, разработчики и аналитики используют различные методы взаимодействия с заказчиками и пользователями. Наиболее популярные методы:

* **Интервью**: один из самых простых и распространённых методов сбора требований, когда аналитики проводят беседы с заказчиками и конечными пользователями, задавая вопросы о том, что должна делать система, и о существующих бизнес-процессах.

Преимущества:

* + Возможность получения подробной информации от ключевых заинтересованных лиц.
  + Гибкость в уточнении требований на месте.

Недостатки:

* + Требует много времени и может быть трудоемким процессом, особенно при большом количестве участников.
* **Опросы и анкеты**: используются, когда нужно получить информацию от большого числа пользователей. Анкеты могут включать как открытые, так и закрытые вопросы.

Преимущества:

* + Можно охватить большое количество участников одновременно.
  + Легкость в сборе и обработке данных.

Недостатки:

* + Меньше гибкости в ответах, так как вопросы должны быть заранее подготовлены.
  + Сложность в углублении и уточнении ответов.
* **Наблюдение**: аналитики наблюдают за тем, как пользователи взаимодействуют с текущими системами или процессами, чтобы лучше понять их потребности.

Преимущества:

* + Позволяет увидеть реальные действия пользователей, а не их интерпретации.
  + Может выявить важные нюансы, которые пользователи могут упустить в интервью.

Недостатки:

* + Может быть сложно наблюдать все процессы.
  + В некоторых случаях может нарушать рабочий процесс.
* **Анализ существующей документации**: если в организации уже используются информационные системы или есть документация по бизнес-процессам, её можно изучить для выявления требований.

Преимущества:

* + Можно получить точную информацию, основанную на реальных данных.
  + Экономия времени, так как не всегда нужно взаимодействовать с пользователями напрямую.

Недостатки:

* + Документация может быть устаревшей или неполной.
  + Некоторые процессы могут быть плохо задокументированы.
* **Рабочие группы и семинары**: используются для обсуждения и уточнения требований в группах, где собираются представители разных отделов компании. Это может быть особенно полезно, если требования затрагивают несколько бизнес-процессов.

Преимущества:

* + Возможность быстрого согласования требований между различными участниками.
  + Более глубокое понимание требований за счёт коллективного обсуждения.

Недостатки:

* + Может быть сложно организовать и управлять процессом обсуждения.
  + Не все участники могут активно принимать участие в дискуссии.

**4. Документирование требований**

После того как требования были собраны, необходимо их правильно задокументировать. Для этого используются различные форматы, такие как:

* **Спецификация требований к программному обеспечению (SRS — Software Requirements Specification)**: официальный документ, который содержит полное описание всех функциональных и нефункциональных требований к системе. Этот документ служит руководством для разработчиков, тестировщиков и менеджеров проекта.

Основные разделы SRS включают:

* + Введение: описание целей проекта, аудитории и контекста.
  + Функциональные требования: описание всех функций системы.
  + Нефункциональные требования: требования к производительности, безопасности, надежности и другим характеристикам системы.
  + Ограничения и допущения: информация о внешних ограничениях, влияющих на систему, таких как бюджеты, сроки и технологии.
* **Диаграммы и модели**: для лучшего понимания требований могут использоваться различные визуальные представления, такие как UML-диаграммы (диаграммы прецедентов, диаграммы классов, диаграммы последовательности) и диаграммы бизнес-процессов (BPMN).

Преимущества:

* + Улучшают понимание требований за счёт визуального представления.
  + Способствуют лучшему общению между техническими специалистами и бизнес-пользователями.

**5. Управление изменениями требований**

Требования к системе могут изменяться в процессе разработки из-за различных факторов, таких как новые бизнес-цели, изменение рыночных условий или улучшенное понимание нужд пользователей. Для эффективного управления изменениями необходимо использовать следующие подходы:

* **Создание гибкой модели разработки**: применение гибких методологий, таких как Agile, которые позволяют адаптироваться к изменяющимся требованиям.
* **Приоритизация изменений**: не все изменения имеют одинаковую важность, поэтому важно правильно расставить приоритеты, чтобы наибольшие ресурсы были направлены на наиболее важные изменения.
* **Документирование изменений**: каждая новая версия требований должна быть задокументирована и согласована с ключевыми заинтересованными лицами.

**Заключение**

Анализ и сбор требований — это фундаментальный этап в процессе разработки информационных систем. От качества этого этапа зависит успех всего проекта, так как неправильно или неполно собранные требования могут привести к проблемам на всех последующих стадиях. Использование различных методов сбора требований, таких как интервью, наблюдение и анализ документации, позволяет получить полную и точную картину того, что нужно от системы. Кроме того, важно учитывать управление изменениями требований, чтобы система могла адаптироваться к новым условиям и потребностям бизнеса.

**Дополнительные материалы:**

1. **Книги**:
   * Wiegers K., Beatty J. "Software Requirements."
   * Robertson S., Robertson J. "Mastering the Requirements Process."
2. **Статьи**:
   * "Effective Requirements Gathering Techniques" (IEEE Software).
   * "The Impact of Poor Requirements on Project Success" (Journal of Information Systems).

**Задания:**

1. Провести интервью с представителем заказчика и составить список функциональных и нефункциональных требований для будущей информационной системы.
2. Составить спецификацию требований для проекта с использованием диаграмм UML и описанием функциональных сценариев.