# Проектирование архитектуры ИИ

UML – унифицированный язык моделирования.

Модель – абстрактное представление реальности в какой-либо форме, предназначенная для рассмотрения определенный аспектов этой реальности.

Жизненный цикл программной системы.

1. Формирование и анализ требований (**Вектор целей**).
2. Анализ предметной области
   1. (Составление адекватного **вектора состояния**)
   2. (as is)
3. Моделирование
   1. (Составление концепции управления процессом разработки)
   2. (Составление концепции работы программы)-(to be)

Разработка концепции работы программы делается итеративно по принципу последовательных приближений моделей от концептуальной модели понятной заказчику, к работе итогового кода понятной программистам (Технологическая модель).

1. Диаграмма вариантов использования (отношений) – самая первая концептуальная модель разрабатывающаяся на основе требований заказчика.

Отношения:

* 1. Ассоциация (актора-человека с действием)
  2. Зависимость (одного действия от другого) – include, extend
  3. Обобщение
  4. Реализация
  5. Прецеденты (CRUD - create, update, delete, read)
  6. Прямоугольник – показывает границы использования программы

Названия вариантов использования (Use case) – Глагол + Существительное

Сценарий использования – это отдельное, независимое действие, которое актер выполняет один раз.

Действия желательно раскрашивать по приоритетам

## Бизнес-моделирование

Система имеет:

1. Структуру
2. Поведение

Модель поведения – описание **алгоритма работы системы**

* + 1. Материальное взаимодействие (**материя**) ->
    2. Перевод в информацию (**мера**)
    3. входная информация (**информация**) ->
    4. алгоритм (**Обработка** **информации**)->
    5. выходная информация (**целевая информация**)->
    6. Код для действий материального носителя (мера)
    7. действие материального носителя (материя)

Диаграмма состояний – объект и функции (состояния) у него. Стрелочки – показывают переходы между состояниями объекта.

Состояния бывают параллельными (вложенными).

Инструменты:

1. Modelio
2. Plant uml

Концептуальная модель:

- use case

- Сценарий

- Диаграмма активностей

- Диаграмма понятий

Логическая модель

- Диаграмма последовательностей

- Диаграмма состояний

- Диаграмма классов

Физическая модель

* Диаграмма компонентов
* Диаграмма развертывания

## Основы программных требований

Уровни требований по Виггерсу:

* Бизнес-требования (зачем заказчику нужна система)
* Пользовательские требования (Что должен делать пользователь)
* Функциональные требования (Что должна делать система)

Требования могут изменяться, поэтому их нужно постоянно соотносить с потребностями.

Бизнес-требования включают в себя:

1. Бизнес-цели
2. Метрики успеха
3. Концепцию
4. Границы и ограничения

Стейкхолдер – источник формирования требований

Источники получения требований

1. Стейкхолдеры (команда разработки тоже относится к стейкхолдерам)
2. Бизнес-процессы
3. Документы
4. Информационные системы

Свойства требований:

1. Полнота требований – требование не требует дополнительной детализации
2. Полнота системы –
3. Корректность – точность описания функциональности

Взаимная корректность или согласованность (непротиворечивость)

1. Осуществимость (выполнимость)
2. Масштабируемость
3. Проверяемость (метрологическая состоятельность)
4. Отслеживаемость

Нефункциональные требования

Внешнее качество (черный ящик):

1. Доступность
2. Удобство установки (удаления)
3. Совместимость
4. Производительность
5. Безопасность
6. Удобство использования
7. Целостность
8. Надежность
9. Устойчивость
10. Защита

Внутреннее качество

1. Эффективность
2. Возможность модификации
3. Переносимость
4. Масштабируемость