

# Лекция 13. Жизненный цикл ПО

Курс «Программирование»

КИТ, 1 семестр

Щукин Александр

Валентинович

# Этапы разработки программного продукта - 1

- Постановка задачи. Оценка осуществимости.
  - сбор информации о задаче,
  - формулировка условия задачи,
  - определение конечных целей решения задачи,
  - определение формы выдачи результатов,
  - описание данных.
- Анализ и исследование задачи, модели.
  - разработка математической модели,
  - разработка структур данных.
- Разработка алгоритмов.
  - выбор метода проектирования алгоритмов,
  - выбор формы записи алгоритмов (блок-схемы, псевдокод),
  - проектирование алгоритмов.

# Этапы разработки программного продукта - 2

- Программирование (кодирование),
  - выбор языка (языков) программирования,
  - уточнение способа организации данных,
  - запись алгоритма на языке программирования,
- Тестирование и отладка:
  - выбор тестов и методов тестирования.
  - синтаксическая отладка,
  - отладка семантики и логической структуры,
  - тестовые расчеты и анализ результатов тестирования.
- Анализ результатов решения задачи и уточнение в случае необходимости математической модели с повторным выполнением шагов 2 – 5.
- Сопровождение программы:
  - составление документации к решенной задаче, к математической модели, к алгоритму, к программе, к набору тестов, к использованию,
  - подготовка к распространению, развертыванию программы,
  - поддержка пользователей,
  - совершенствование программы.

# Реинжиниринг



# Жизненный цикл программного продукта

- Модель ЖЦ ИС – это структура, определяющая последовательность процессов, действий и задач, выполняемых на протяжении ЖЦ ИС, а также взаимосвязи между ними.
- Оценка рисков продолжения работ

# Водопадная (каскадная) модель

- Разбиение разработки на этапы
- Последовательная организация работ
- Выпуск полного комплекта документов при переходе на следующий этап

# Водопадная (каскадная) модель



# Недостатки водопадной модели

- существенная задержка получения результатов;
- необходимость возврата на предыдущие этапы;
- сложность распараллеливания работ по проекту;
- информационная перенасыщенность каждого этапа;
- сложность управления проектом;
- высокий уровень риска и ненадежности инвестиций.

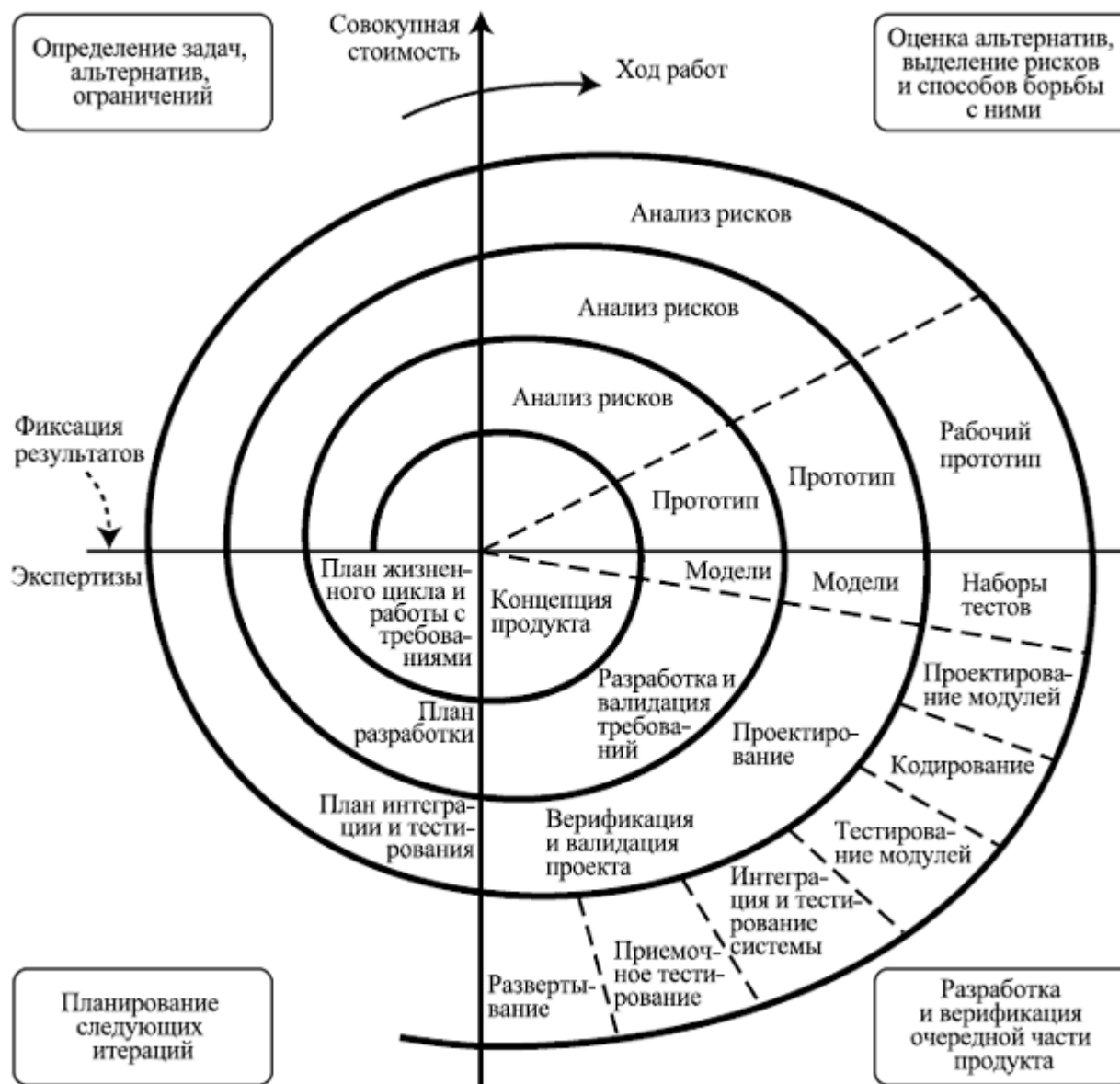


# Спиральная модель

- Итерационный процесс работ
- Активное использование прототипов
- Система версий
- Постепенная конкретизация
- Постепенная интеграция компонентов в сложных системах
- Снижение уровня рисков
- Упрощение внесения тактических изменений

# Спиральная модель

(с) Александр В. Щукин, СПбПУ, ИКНТ, КИТ



# Недостатки спиральной модели

- Определение момента перехода на следующий этап
- Бесконечное совершенствование.

# Agile-методология

- Agile software development (гибкая методология разработки)
- Agile-методы — обобщающий термин для целого ряда подходов и практик, основанных на ценностях Манифеста гибкой разработки программного обеспечения и 12 принципах
- Работа в небольших группах
- Серия коротких циклов
- Уменьшение объема письменной документации

# Agile Manifesto. Идеи

- люди и взаимодействие важнее процессов и инструментов;
- работающий продукт важнее исчерпывающей документации;
- сотрудничество с заказчиком важнее согласования условий контракта;
- готовность к изменениям важнее следования первоначальному плану.

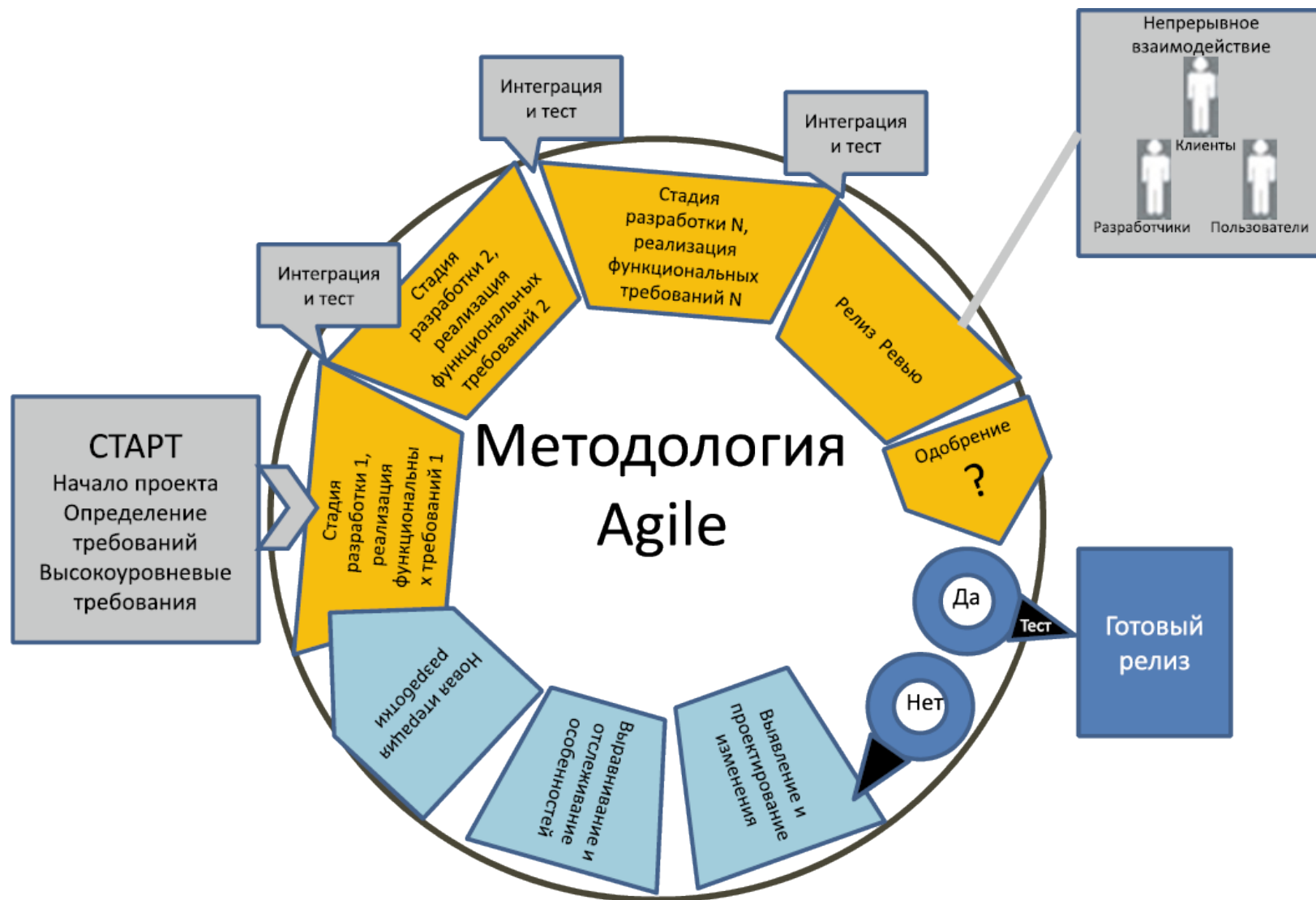
# Agile Manifesto. Принципы

- удовлетворение клиента за счёт ранней и бесперебойной поставки ценного программного обеспечения;
- приветствие изменений требований даже в конце разработки (это может повысить конкурентоспособность полученного продукта);
- частая поставка рабочего программного обеспечения (каждый месяц или неделю или ещё чаще);
- тесное, ежедневное общение заказчика с разработчиками на протяжении всего проекта;
- проектом занимаются мотивированные личности, которые обеспечены нужными условиями работы, поддержкой и доверием;

# Agile Manifesto. Принципы (2)

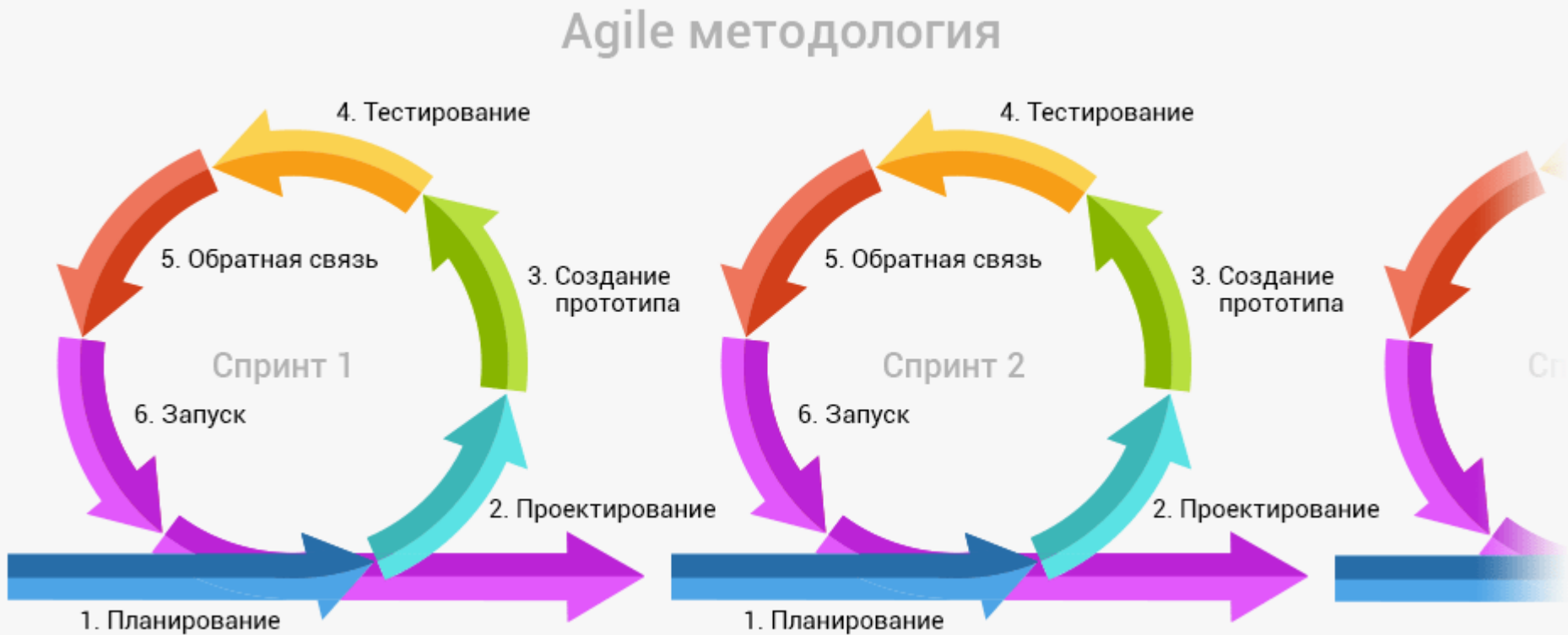
- рекомендуемый метод передачи информации — личный разговор (лицом к лицу);
- работающее программное обеспечение — лучший измеритель прогресса;
- спонсоры, разработчики и пользователи должны иметь возможность поддерживать постоянный темп на неопределённый срок;
- постоянное внимание улучшению технического мастерства и удобному дизайну;
- простота — искусство не делать лишней работы;
- лучшие технические требования, дизайн и архитектура получаются у самоорганизованной команды;
- постоянная адаптация к изменяющимся обстоятельствам. Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.

# Agile-методология





# Agile-методология (2)



# Этапы разработки программного продукта (другой взгляд)



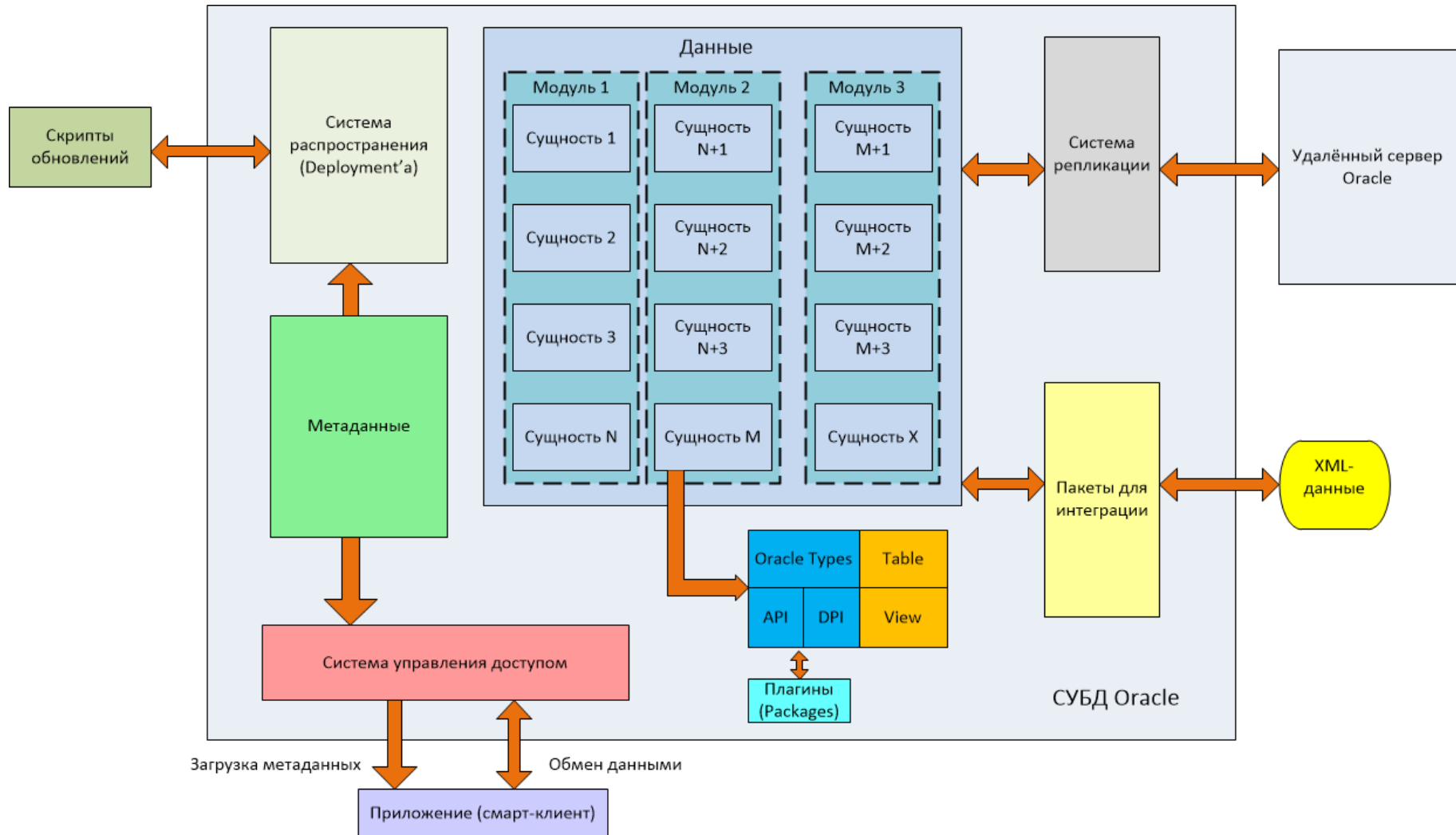
# Единая система программной документации

- Техническое задание
- Программа и методика испытаний
- Руководство системного программиста
- Руководство пользователя
- Руководство программиста. SDK. API

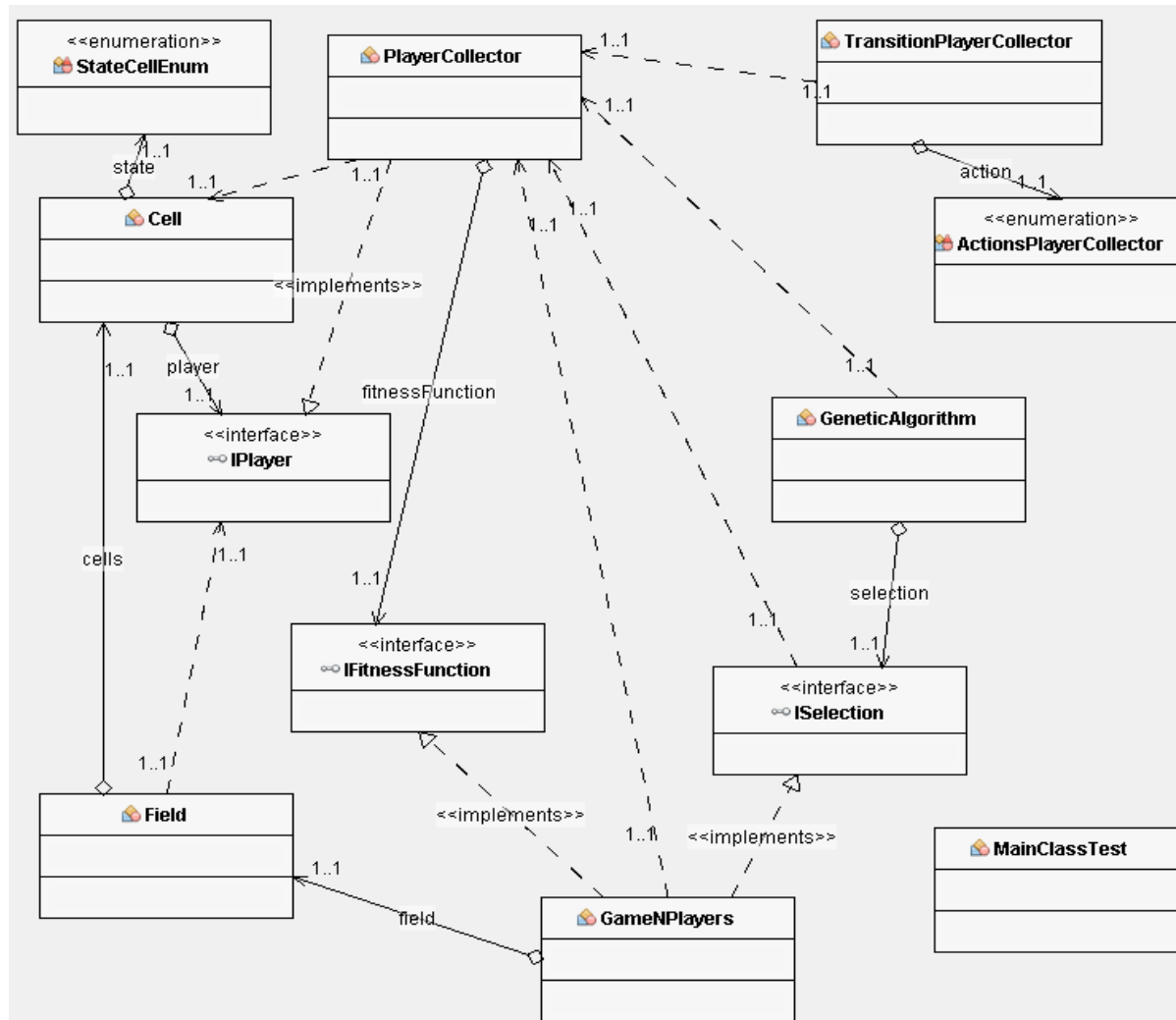
## Спецификации ПО

- Структурная модель – схематическое представление структуры информационной системы
- Функциональная, поведенческая модель – взаимодействие системы или ее отдельных компонент и (возможно) внешнего окружения
- Модель данных – структуры для представления данных
- Объектная модель – представление классов в терминах ООП

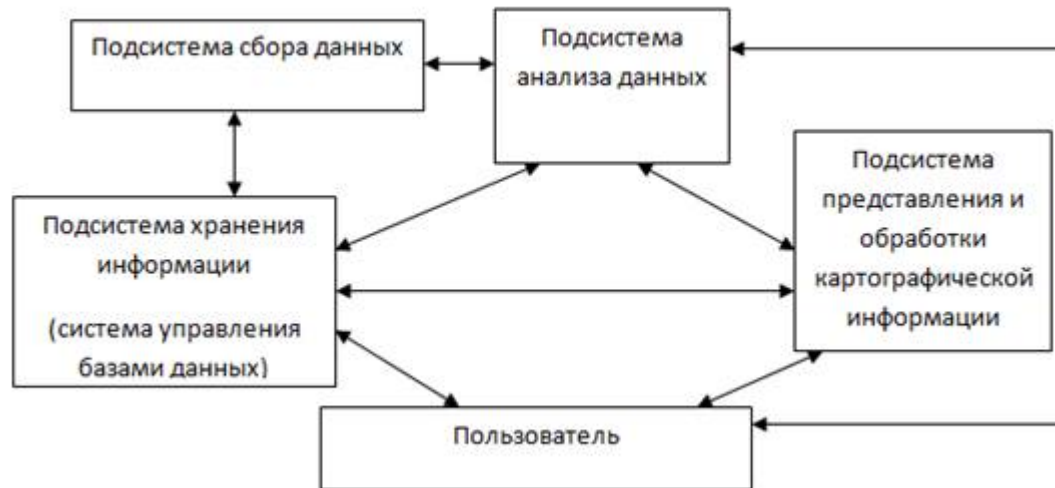
# Функциональная модель



# Объектная модель



# Функциональная модель



# Структура базы данных

