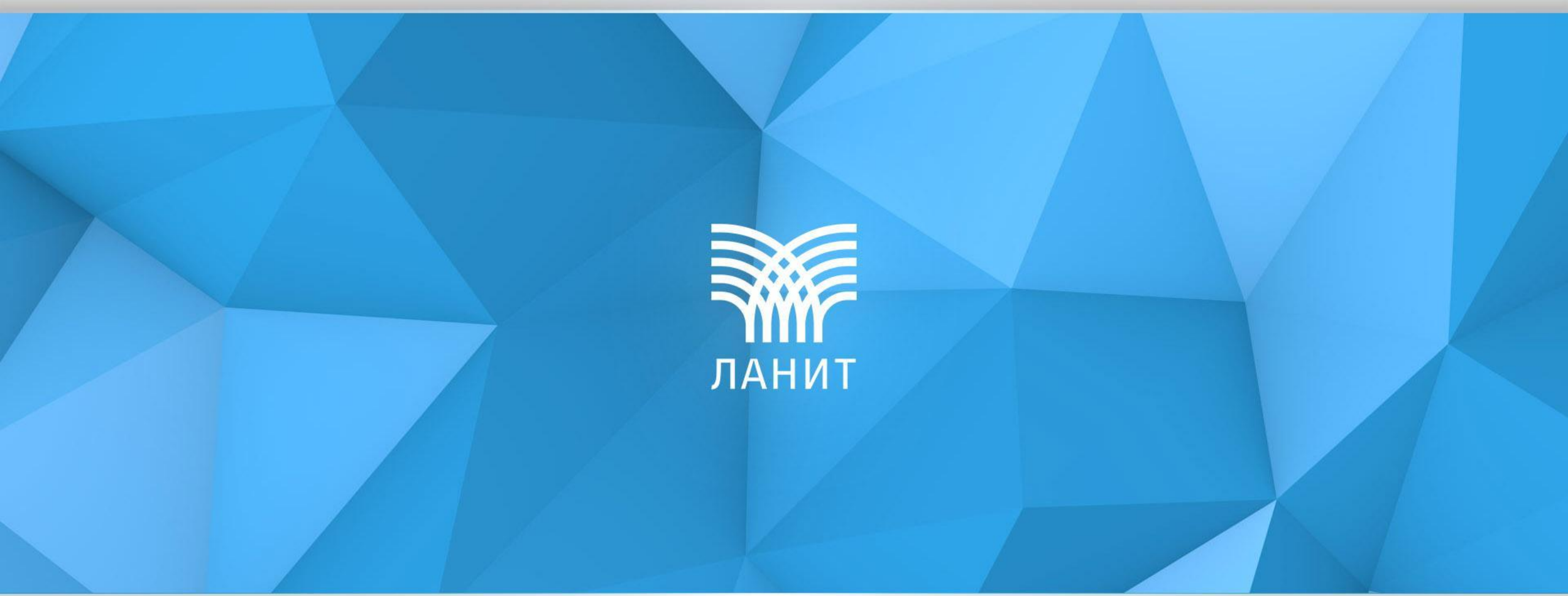


Введение в функциональное тестирование

Москва, 2018

Содержание

Введение

Тестирование ПО, равно как и разработка ПО, является наиболее востребованной профессией в IT сфере. Такие организации как the British Computer Society или the International Software Testing Qualifications Board (ISTQB) разработали и поддерживают сертификационные уровни тестирования.

Тестировщики должны обладать знаниями, техническими навыками и определенными талантами для того, чтобы в сжатые сроки протестировать продукт таким образом, чтобы он максимально полно соответствовал потребностям Заказчика.

Тестирование не интуитивный процесс, тестировщик **ДОЛЖЕН** знать, как это делается.

Основные понятия и определения (1 из 4)

- **Тестирование** – процесс, помогающий определить корректность, полноту и качество разработанного программного обеспечения. Вместе с тем, тестирование никогда не может полностью установить корректность программы. Только процесс формальной проверки может доказать, что дефекты отсутствуют.
- **Тестирование** (ISTQB) – процесс, содержащий в себе все активности жизненного цикла, как динамические, так и статические, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результатов работ с целью определить, что они соответствуют описанным требованиям, показать, что они подходят для заявленных целей и для определения дефектов.
- **Качество** – способность программы делать то, что ждет от нее пользователь.
- **Надежность** – вероятность того, что программа будет работать без ошибок определенный промежуток времени.
- **Основная цель тестирования:** Выпустить на рынок качественный продукт

Качество ПО. Зачем?

1. Обеспечение информацией программистов, которую они могут использовать для предотвращения ошибок;

2. Обеспечение менеджеров информацией, которая необходима им для разумной оценки риска при использовании объекта;

3. Создание объекта максимально свободного от ошибок;

4. Создание проекта поддающегося тестированию, т.е. проекта, который можно легко проверить на соответствие, на искаженность и который будет легко сопровождать;

5. Проверка искаженности объекта с помощью как сформулированных, так и не сформулированных требований. Это еще называют «взломом программного обеспечения»;

6. Проверить соответствие объекта (убедиться в его действенности), т.е. показать, что он работает правильно.

Quality Control vs Quality Assurance

Основные понятия и определения (2 из 4)

- **Тестируемость** – степень, до которой могут быть запланированы объективность и реализуемость тестирования, проверяющего соответствие требованию.
- **Ошибка** – действие человека, которое приводит к неправильному результату.
- **Дефект** (fault) – возможная причина отказа. Изъян в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию, например неверный оператор или определение данных. Дефект, обнаруженный во время выполнения, может привести к отказам компонента или системы.

Основные понятия и определения (3 из 4)

- **Среда тестирования** (test environment) – инфраструктура для подготовки и проведения тестирования и интерпретации результатов.
- **Покрытие** (test coverage) – часть функционала, тестируемая данным набором проверок.

Основные понятия и определения (4 из 4)

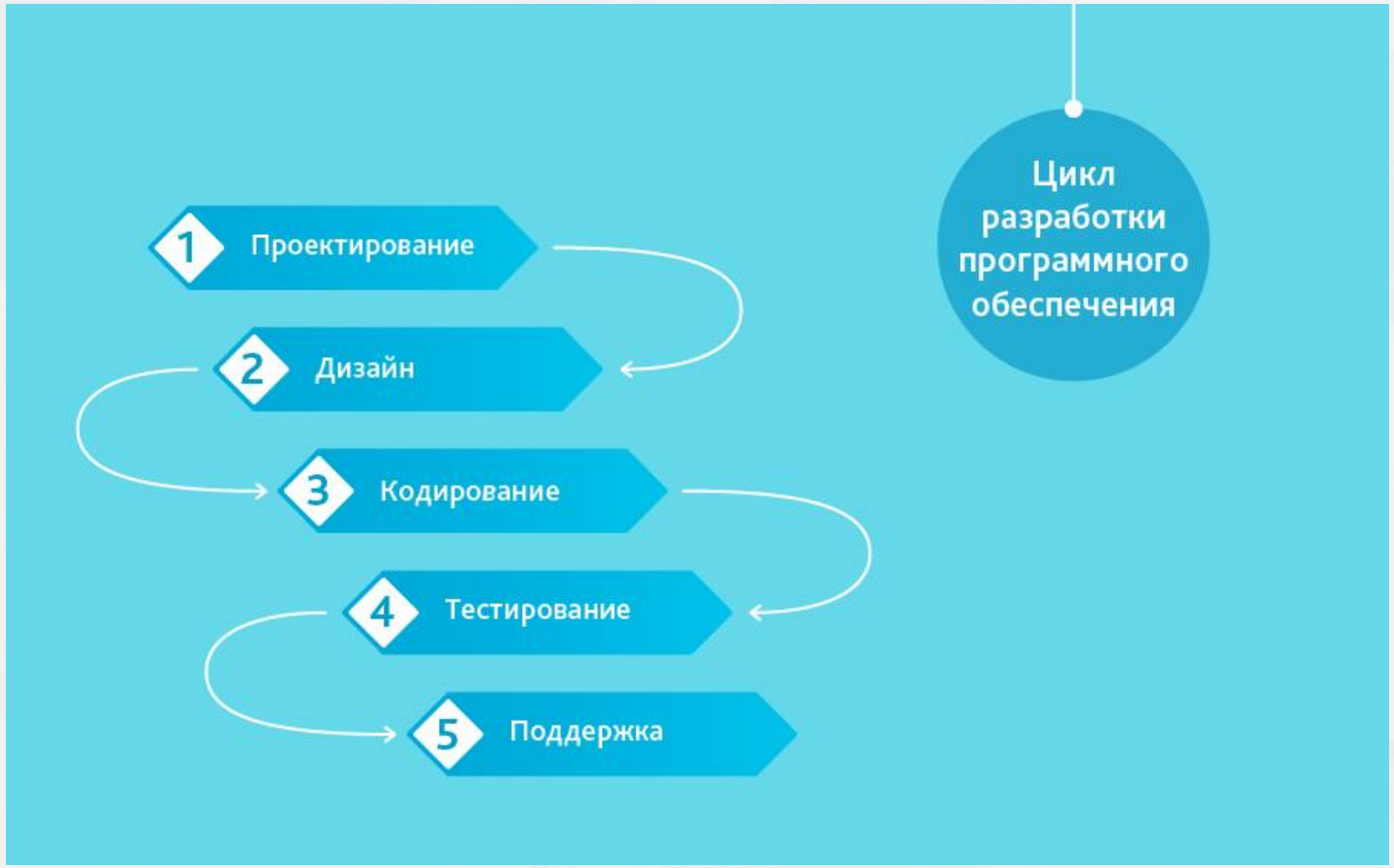
- **Валидация** (validation) – проверка того, что программа делает то, что нужно.
- **Верификация** (verification) – проверка того, что программа делает все правильно.
- **Тест кейс** – выполняемый тест с конкретными входными данными, начальными условиями и ожидаемым результатом.

Методология разработки ПО – это шаблон, определяющий взаимодействие различных составляющих процесса получения версии какой-либо программы.

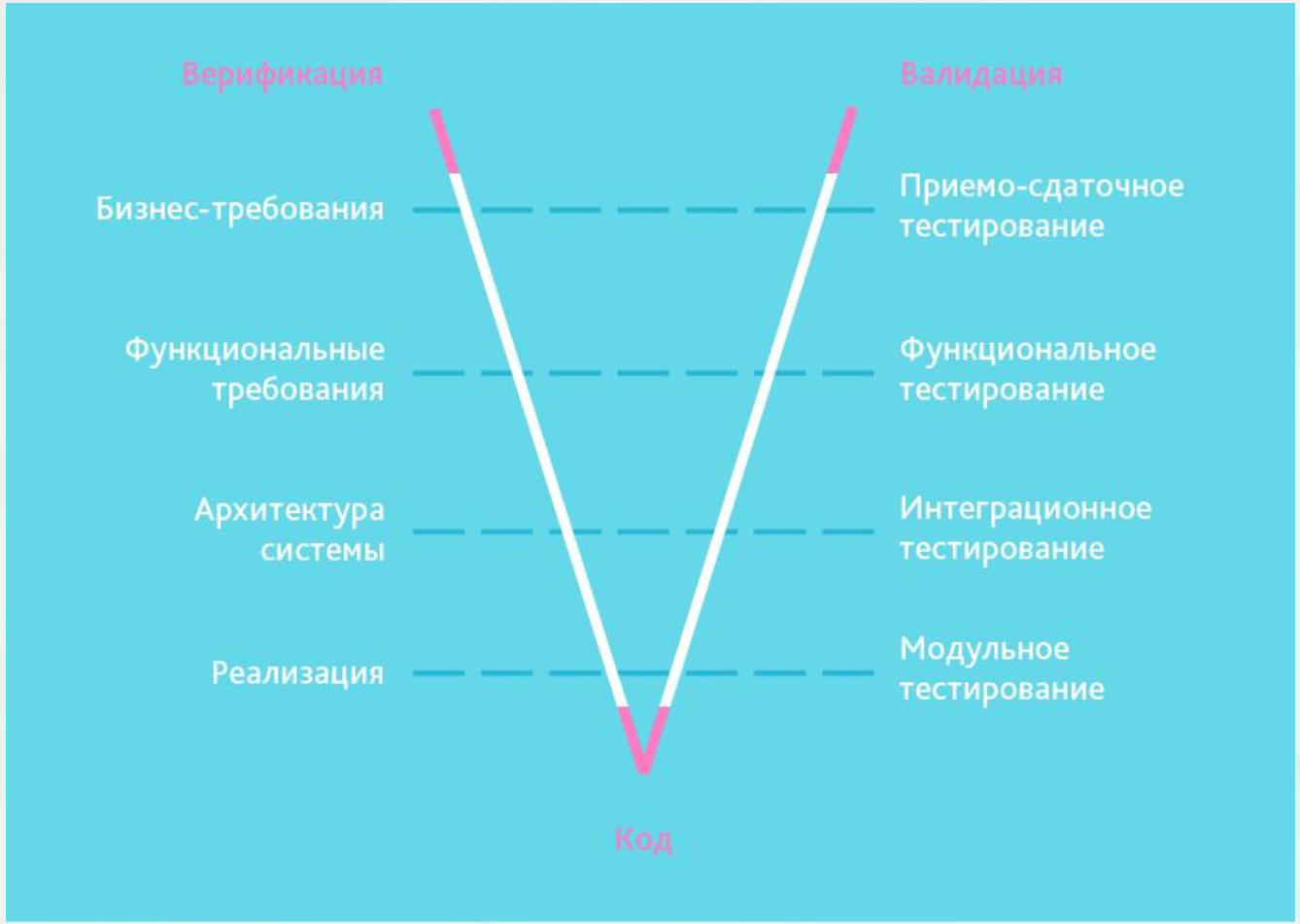
Методология разработки описывает как мы будем разрабатывать продукт.

- Модель «водопада»;
- V-образная модель;
- Инкрементальная модель;
- Спиральная модель;
- Гибкая модель разработки;
- Итеративная модель.

Модель «Водопад» / Каскадная



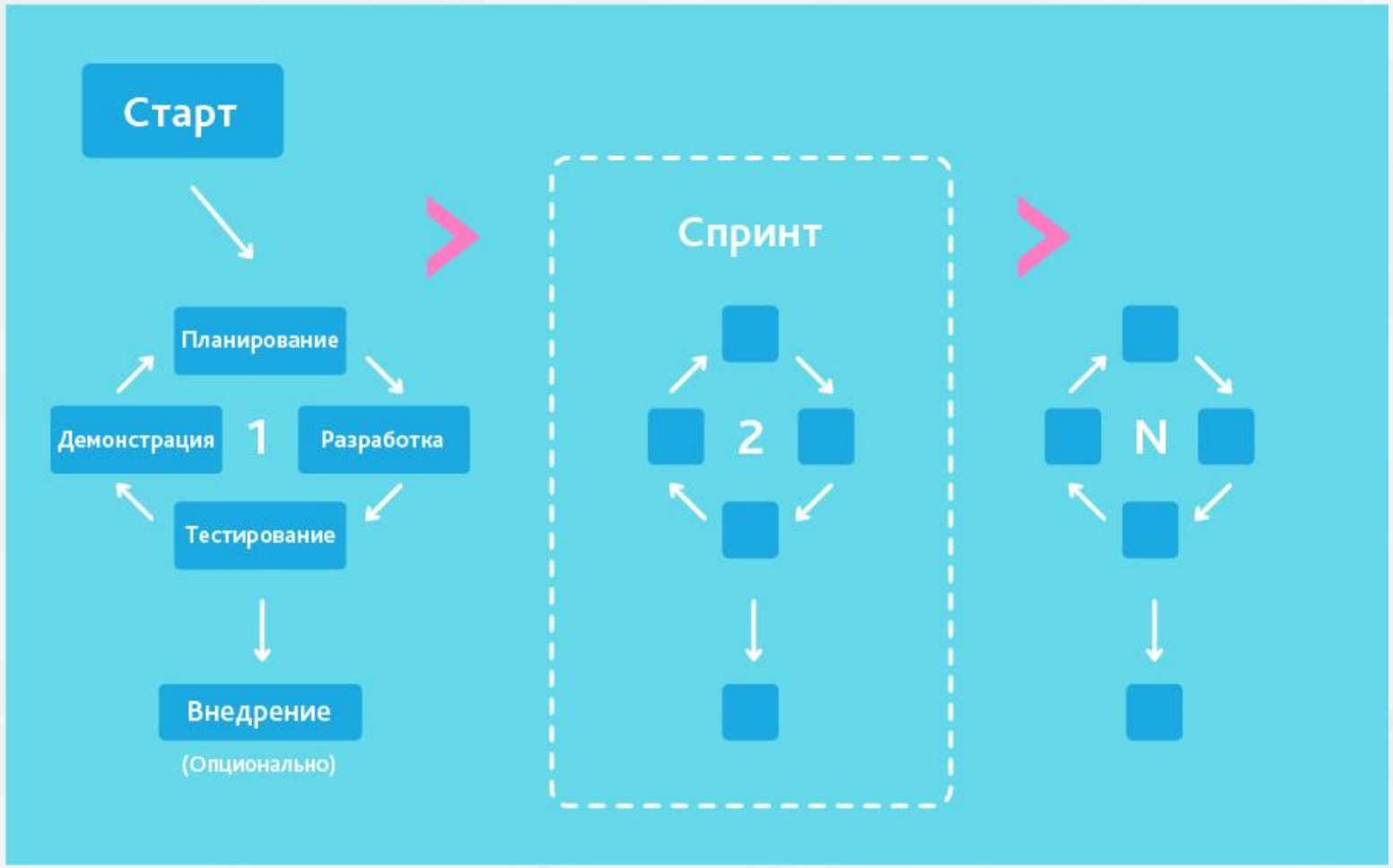
V-образная модель



Инкрементальная модель

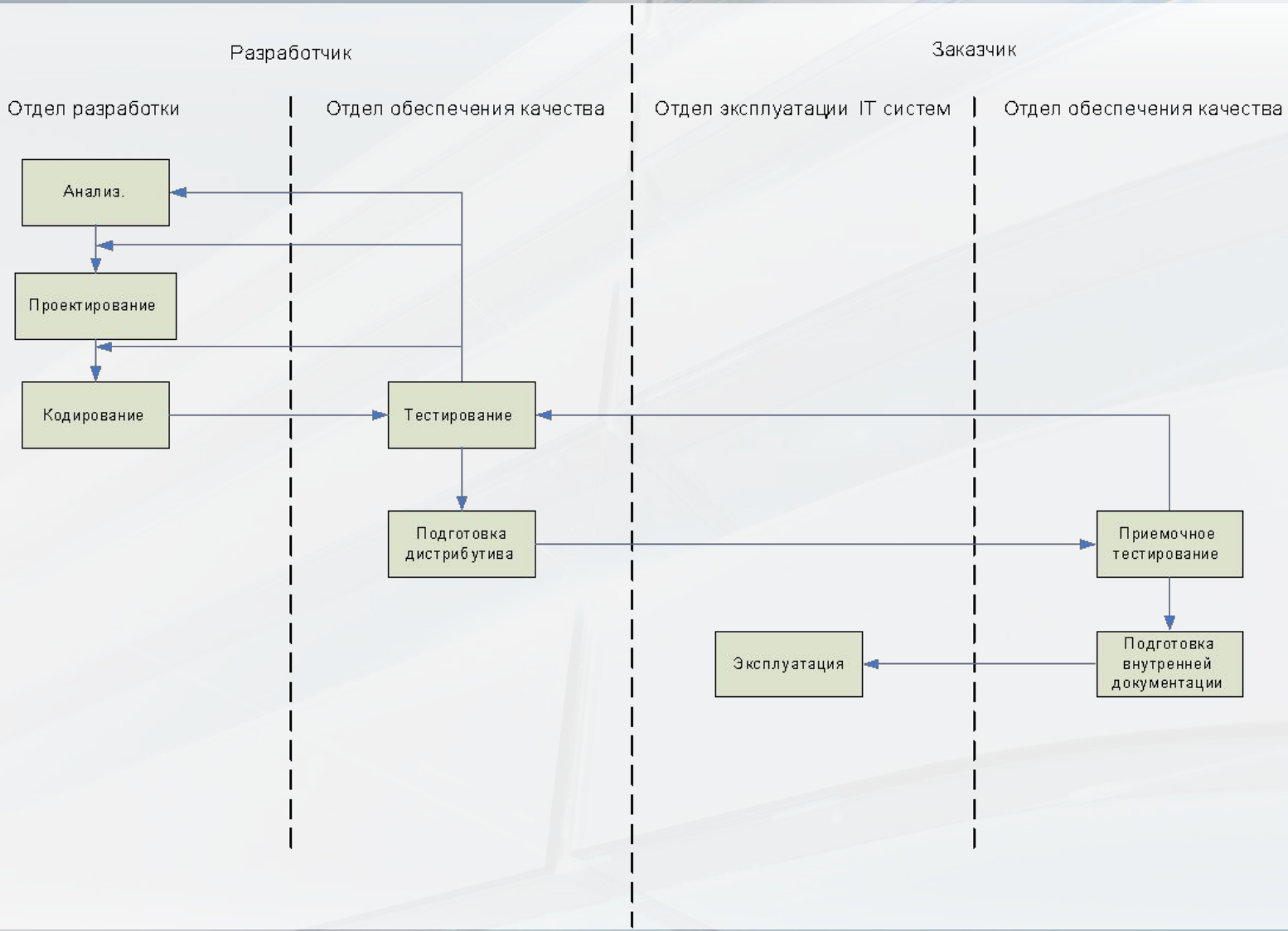
Спиральная модель

Гибкая методология разработки



Итеративная модель

Типовой пример установки заказного ПО



Место тестирования в ЖЦПО

7 Принципов тестирования

Принцип 1 – Тестирование демонстрирует наличие дефектов

Принцип 2 – Исчерпывающее тестирование недостижимо

Принцип 3 – Раннее тестирование

Принцип 4 – Скопление дефектов

Принцип 5 – Парадокс пестицида

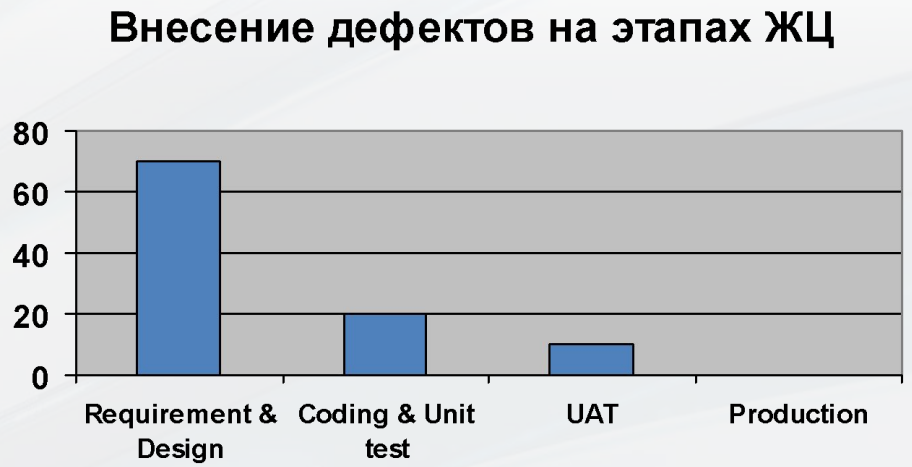
Принцип 6 – Тестирование зависит от контекста

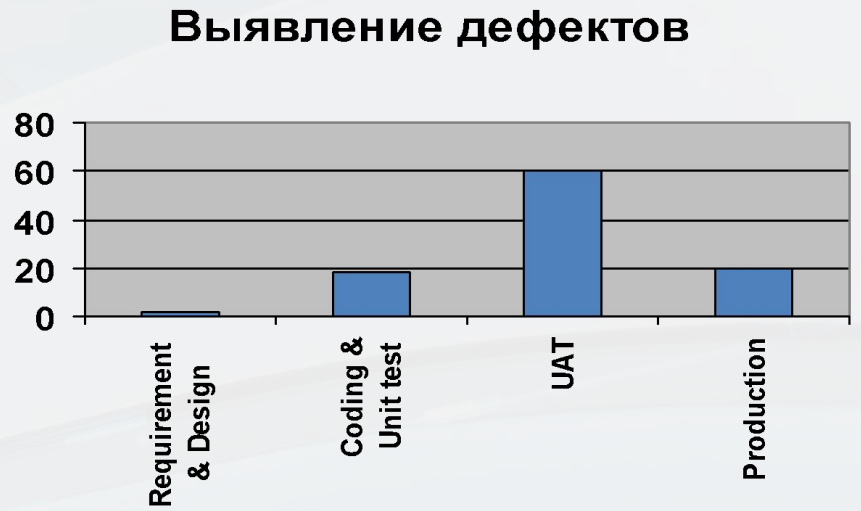
Принцип 7 – Заблуждение об отсутствии ошибок.

Важность тестирования

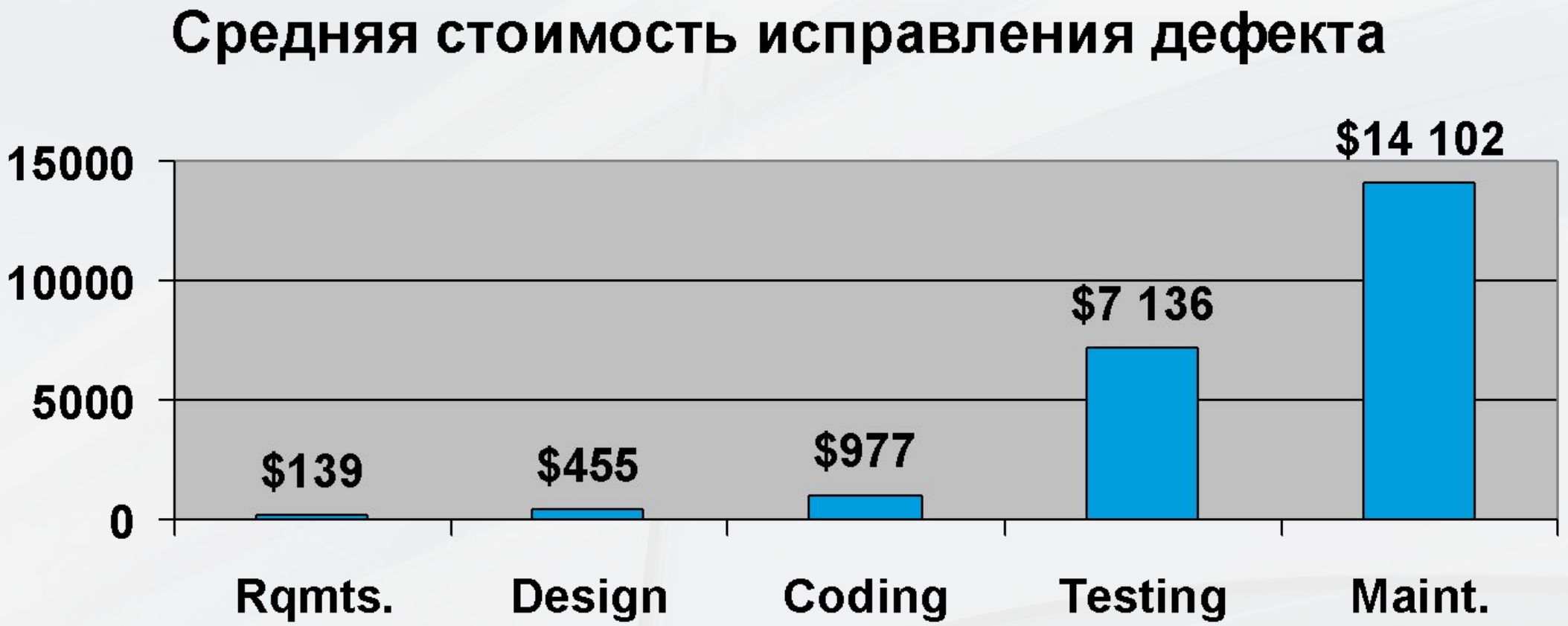
- Большинство дефектов вносятся на этапе сбора требований и разработки

- … а обнаруживаются на стадии приемочных испытаний.





Важность тестирования



Стоимость исправления ошибки возрастает с каждым этапом ЖЦ

Принципы тестирования (Майерс)

- Описание предполагаемых значений выходных данных или результатов должно быть необходимой частью тестового набора
- Следует избегать тестирования программы ее автором
- Необходимо досконально изучать результаты применения каждого теста
- Тесты для неправильных и непредусмотренных входных данных следует разрабатывать так же тщательно, как для правильных и предусмотренных
- Необходимо проверять не только, делает ли программа то, для чего она предназначена, но и ни делает ли она то, что не должна делать
- Не следует выбрасывать тесты, даже если программа уже не нужна
- Нельзя планировать тестирование в предположении, что ошибки не будут обнаружены
- Вероятность наличия необнаруженных ошибок в части программы пропорциональна числу ошибок, уже обнаруженных в этой части
- Тестирование – процесс творческий. Вполне вероятно, что для тестирования большой программы требуется больший творческий потенциал, чем для ее проектирования

Задачи и виды тестирования

- **Тестирование нового функционала** (functionality)
- **Регрессионное тестирование** (regression)
- **Интеграционное тестирование** (integration)

Тестирование нового функционала

- Проверка соответствия поведения системы/подсистемы и т.д. спецификациям функциональных требований (requirements specification)

Регрессионное тестирование

- Тестирование уже протестированной программы, проводящееся после модификации для уверенности в том, что процесс модификации не внес или не активизировал ошибки в областях, не подвергавшихся изменениям.
- Проводится после изменений в коде программного продукта или его окружении.
- При большой истории все проверки выполнить невозможно, поэтому, как правило, они выполняются выборочно.

Smoke тестирование

- Выборка из общего числа запланированных тестовых сценариев, покрывающая основную функциональность компонента или системы.
- Проводится с целью удостовериться, что базовые функции программы в целом работают корректно, без углубления в детали.
- Ежедневная сборка и тест "на дым" являются передовыми практическими методами.

Интеграционное тестирование

Тестирование, выполняемое для обнаружения дефектов в интерфейсах и во взаимодействии между интегрированными компонентами или системами.

1. Компонентное/Модульное интеграционное тестирование;
2. Системное интеграционное тестирование.

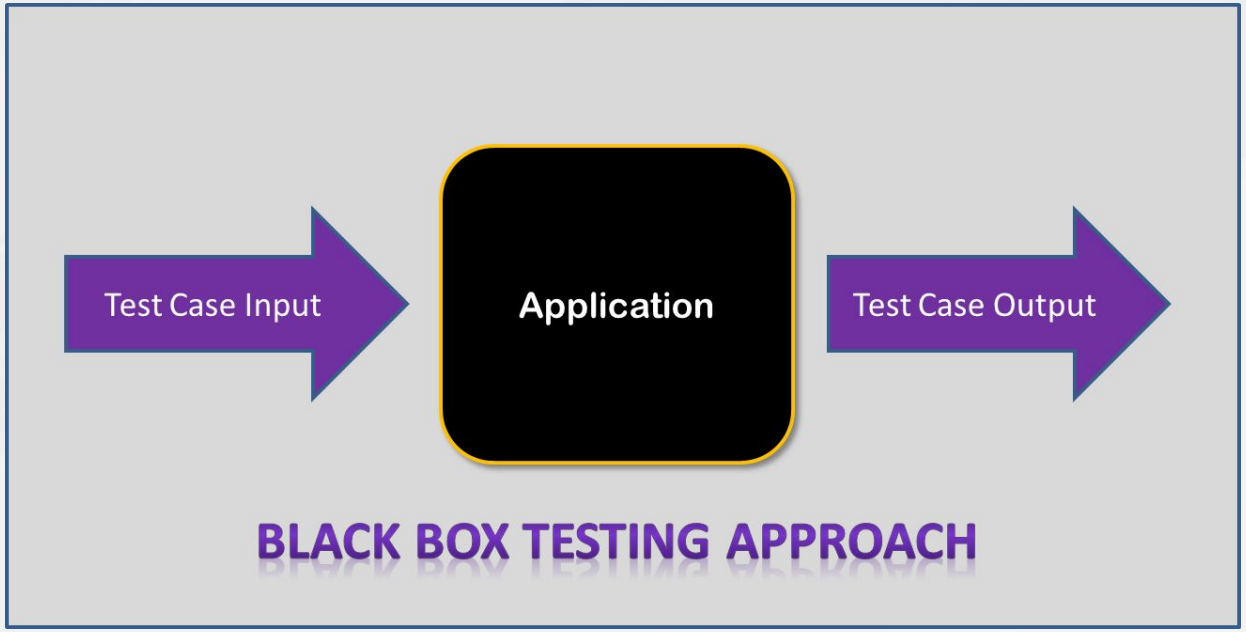
Методы тестирования (1 из 4)

- **Белый ящик** – тестирование на основе анализа структуры кода
- **Серый ящик** – промежуточный уровень, на котором определен интерфейс между компонентам.
- **Черный ящик** – поведенческое тестирование, глобальное представление, основанное на спецификации.

Методы тестирования (2 из 4). «Чёрный» ящик

Данный метод включает в себя следующие процедуры:

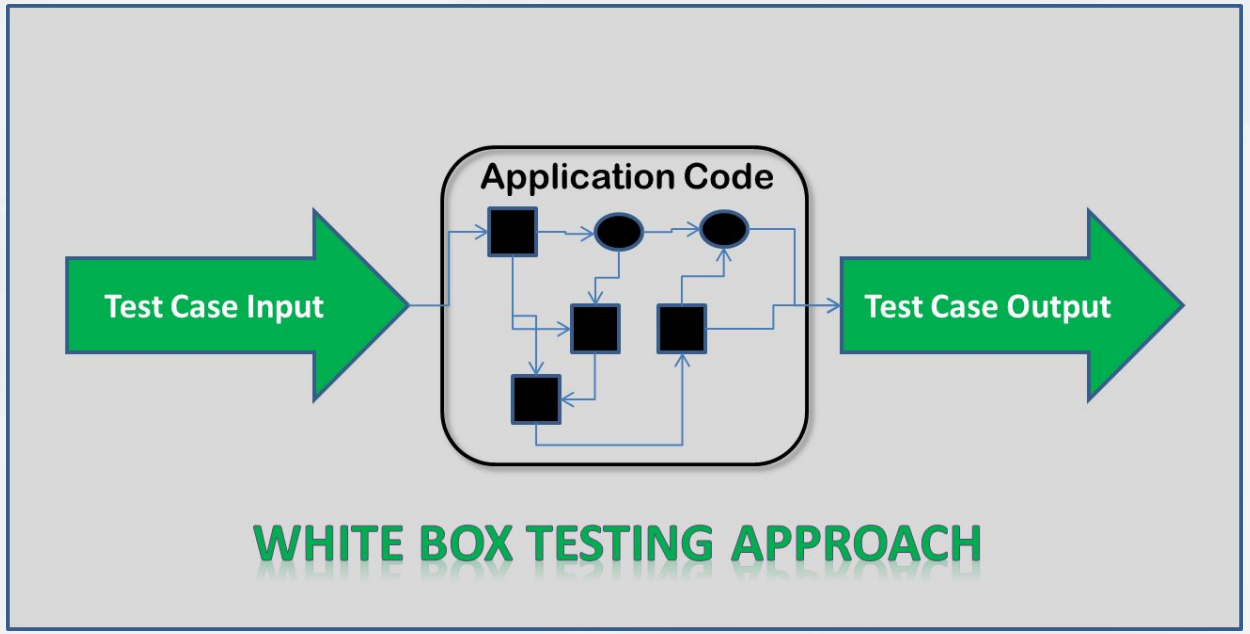
- анализ потоков
- таблица решений
- функциональный анализ
- граничные значения
- история ошибок
- целостность данных
- диаграмма переходов состояний



Методы тестирования (3 из 4). «Белый» ящик

Данный метод включает в себя следующие процедуры:

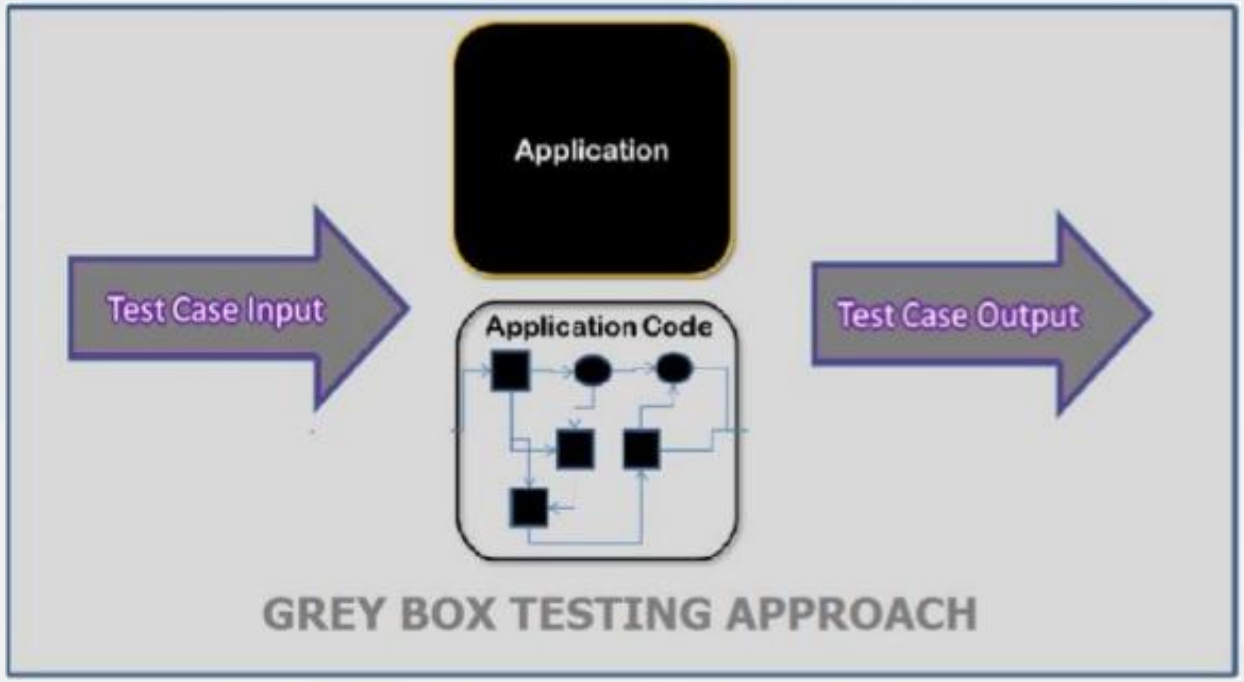
- анализ структуры программы
- покрытие операторов
- покрытие ветвей



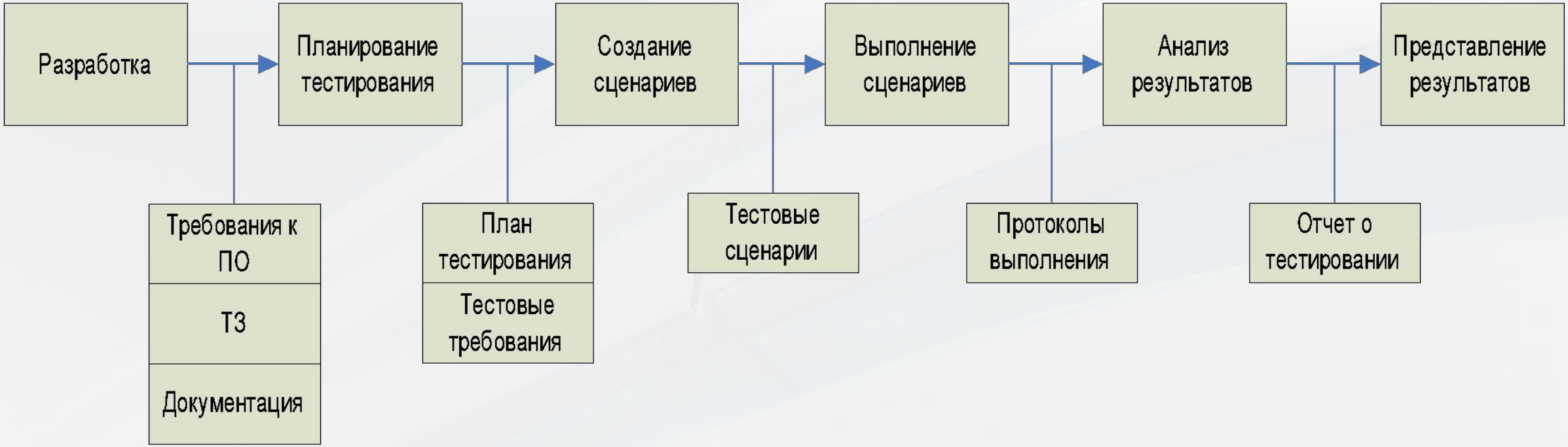
Методы тестирования (4 из 4). «Серый» ящик

Данный метод включает в себя следующие процедуры:

- анализ архитектуры
- проверка навигации
- проверка интерфейсов
- модель событий



Ход тестирования



- Планирование
- Выполнение
- Анализ результатов
- Представление результатов

Задание по лекции 1

- Опишите что такое качество продукта. Приведите примеры качественного и некачественного продукта;
- Что такое валидация и верификация? Опишите с примерами в чем различие между ними;
- Какие бывают модели разработки?
- В каких случаях предпочтительнее использовать ту или иную модель разработки?
- В каком случае нельзя использовать гибкую модель разработки? Приведите пример такого случая.

Контакты

129075, г. Москва,

Мурманский проезд, д. 14, к. 1

Тел./факс: +7 (495) 967 66 50

lanit@lanit.ru


ЛАНИТ