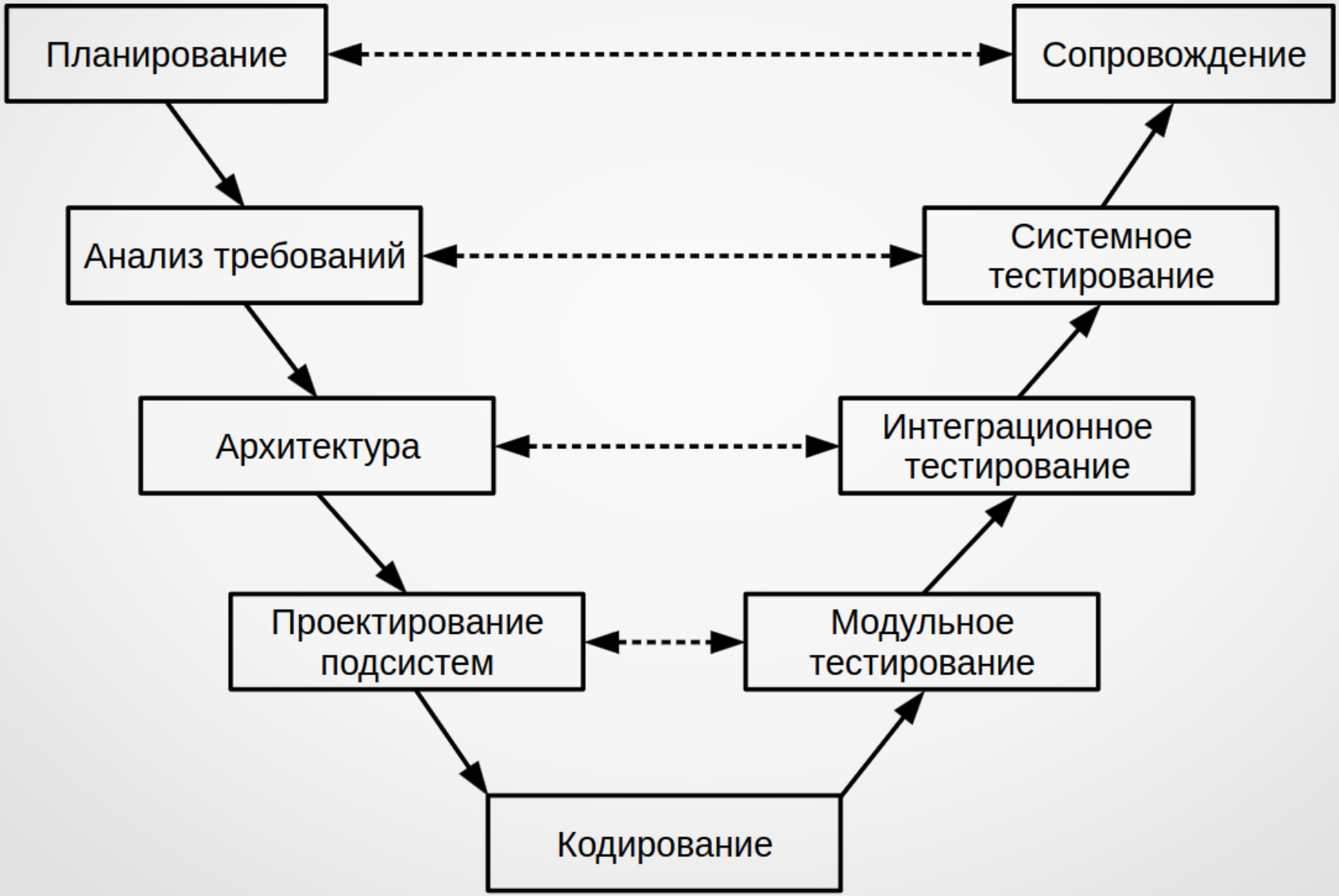
Основы разработки ПО

Классификации тестирования

Кулаков Кирилл Александрович

V-модель разработки ПО



Модульное тестирование

- Структурное тестирование (белый ящик)
- Выполняется самим разработчиком
- Поиск дефектов
  - алгоритмические ошибки
  - ошибки кодирования алгоритмов
  - выполнение условных и циклических операторов
  - использование переменных и ресурсов
- Использование заглушек, эмуляторов и других вспомогательных инструментов, заменяющих полностью или частично реальные компоненты ПО

Модульное тестирование

- Преимущества
  - Возможность протестировать часть программы, не ожидая готовности остальных частей
  - Раннее обнаружение дефектов
  - Программисты обнаруживают и мгновенно исправляют проблемы. Упрощенная отладка
  - Лучшее структурное покрытие кода
  - Модульное тестирование экономичнее других этапов тестирования
  - Упрощенная интеграция

Модульное тестирование

- Недостатки
  - Время от времени требуется реализовывать заглушки и драйвера
  - Модульное тестирование основано, в первую очередь, на написанном коде. Поэтому, если что-то было пропущено, модульное тестирование этого не покажет
  - Изменение кода — изменение заглушек и драйверов
  - Функциональные ошибки модульное тестирование может не показать

Интеграционное тестирование

- Смешанная модель разработки (Серый ящик)
- Поиск ошибок
  - трактовка данных
  - реализация интерфейса взаимодействия
  - совместимость
- Схема интеграции
  - восходящая (от малого к большому)
  - нисходящая (подключение к основному модулю)
  - смешанная (сборка блоков и подключение к основному модулю)

Интеграционное тестирование

	Восходящее	Нисходящее
Преимущества	• Возможность ранней проверки корректности низкоуровневого поведения • Не требуется написание заглушек • Просто определить требования ко входам/выходам модулей	• Возможность ранней проверки корректности высокоуровневого поведения • Модули могут добавляться по одному, независимо друг от друга • Не требуется разработка множества драйверов • Можно разрабатывать систему как в глубину, так и в ширину
Недостатки	• Отложенная проверка высокоуровневого поведения • Требуется разработка драйверов • При замене драйвера на модуль высокого уровня может произойти "мини-Большой Взрыв" числа найденных ошибок	• Отложенная проверка низкоуровневого поведения • Требуется разработка "заглушек" • Крайне сложно корректно сформулировать требования ко входам/выходам частичной системы

Переходы между состояниями

- Переход м/у состояниями в следствие действий пользователя или внешнего окружения
- Переходов может быть очень много!
  - Протестируйте все наиболее вероятные последовательности действий пользователей.
  - Если можно предположить, что действия пользователя в одном режиме могут воздействовать на представление данных или набор предоставляемых программой возможностей в другом режиме, протестируйте эту зависимость.
  - Кроме проведения самых необходимых тестов — из тех, что описаны выше, — стоит поработать с программой в произвольном режиме, случайным образом выбирая путь ее выполнения.
- Построение схем меню и форм

Интеграционное тестирование

- Преимущества
  - Большая стабильность по сравнению с тестированием графического пользовательского интерфейса
  - Положительно влияет на внутренний дизайн программы
  - Ранняя и более легкая локализация дефектов интерфейса на стадии системного тестирования
- Недостатки
  - Тестировщик должен читать код, а временами и писать его

Регрессионное тестирование

- Исправление ошибки зачастую порождает новые ошибки
  - Проверка исправления ошибки
  - Поиск связанных ошибок
  - Проверка остальной части программы
- Использование библиотеки регрессионных тестов
  - Удаление тестов эквивалентных другим тестам библиотеки.
  - Уменьшение количества тестов, объектом которых является уже исправленная ошибка
  - Комбинации тестов
  - Автоматизация тестирования
  - Периодическое выполнение

Системное тестирование

- Функциональное тестирование (черный ящик)
- Проверка на соответствие стандартам и требованиям
- Оценка характеристик качества ПО
  - устойчивость
  - надежность
  - безопасность
  - производительность
- Приемочное тестирование
- Демонстрация
- Тестирование пользователями

Системное тестирование

- Примеры тестов
  - Тестирование наличия заявленной функциональности
  - Тестирование производительности
  - Нагрузочное или стрессовое тестирование;
  - Тестирование конфигурации;
  - Тестирование безопасности;
  - Тестирование надежности и восстановления после сбоев;
  - Тестирование удобства использования

Анализ чувствительности

- Поиск тестов, вызывающих наибольшие возмущения
  - Общее представление о поведении функции, на основе ее значений для ряда параметров, располагающихся вдоль всей области определения
  - Поиск участков области определения, на которых небольшие изменения аргументов вызывают значительные скачки результирующих значений
- Оценка погрешностей получаемых значений (операции с плавающей запятой)
- Использование теории вероятности для тестирования мат.функций
  - Пример: сколько тестов надо если в меню 3 ссылки на формы, каждая форма имеет по 4 кнопки, при нажатии на кнопку заполняется отчет, тип которого зависит от флажка?

Нагрузочные испытания

- Нагрузочное испытание — один из видов пограничного тестирования
- Проверка максимально допустимых ресурсов (размер файлов, количество соединений, устройств и т.п.)
- Условия гонок
  - Проверка работы в нагруженном и ненагруженном режиме
  - Воздействия на приложение во время работы
  - Проверка на быстрых и медленных машинах

Пользовательское тестирование

- Пользовательское тестирование
- Альфа-тестирование (в условиях компании-разработчика)
- Бета-тестирование (в условиях компании-заказчика)
- Произвольное использование (bag bush)
- Тестирование экспертом
- Парное тестирование
- Использование внутри компании (Eat your own dogfood)
- Тестирование локализации