

Тестирование программного обеспечения

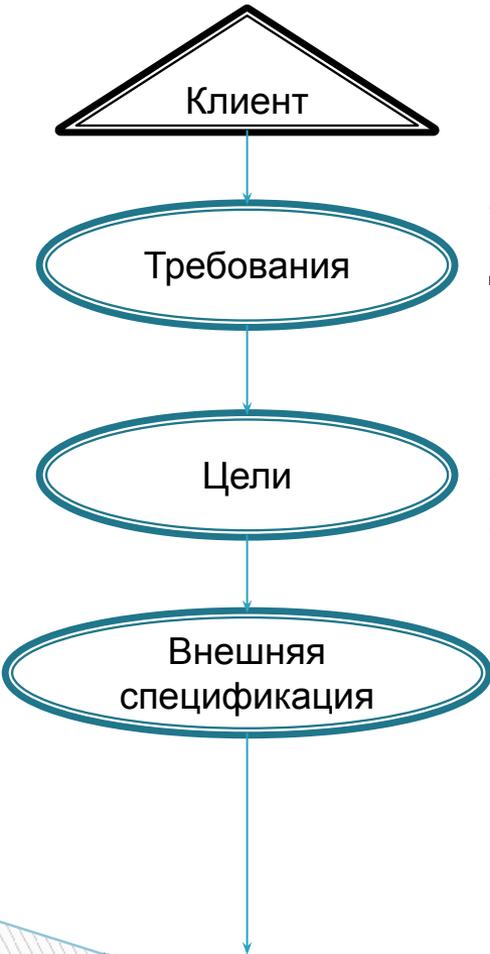
Высокоуровневое тестирование

Высокоуровневое тестирование

- ▣ Завершение модульного тестирования не означает окончания тестирования программы;
- ▣ В ПО содержится ошибка, если оно не оправдывает разумных ожиданий пользователя.

Разработка ПО в значительной степени представляет собой процесс передачи информации о том, что собой должна представлять конечная программа, и трансляции этой информации из одной формы в другую, от общего – к частному.

Процесс разработки ПО

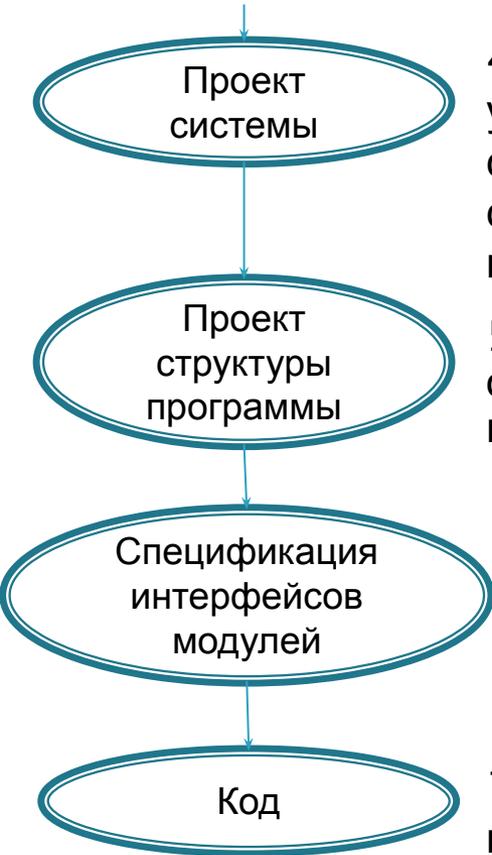


1) Перевод потребностей клиента в документированные требования. Эти требования представляют общие цели, которые должен реализовать разрабатываемый продукт;

2) Перевод в формулировки конкретных целей путём оценки их осуществимости, а также временных и денежных затрат; при этом устраняются противоречия между требованиями и устанавливаются необходимые приоритеты и компромиссы;

3) Перевод целей в точное описание продукта, причем сам продукт рассматривается как “чёрный ящик” и учитываются лишь его интерфейсы и взаимодействия с конечным пользователем. Это описание называется **внешней спецификацией**.

Процесс разработки ПО



4) Если продукт представляет собой систему(ОС, система управления полётами или др.), а не обычное приложение, то следующий шаг – проектирование системы. На данном этапе систему разбивают на отдельные программы, компоненты или подсистемы и описывают их интерфейсы.

5) Проектирование структуры программы или программ путём определения функции каждого модуля, иерархической структуры модулей, а также интерфейсов модулей.

6) Разработка точной спецификации, определяющей функции и интерфейсы каждого модуля.

7) Перевод, путём выполнения одного или нескольких вспомогательных этапов, спецификации интерфейса каждого модуля в исходный код соответствующего алгоритма.

Процесс разработки ПО

Эти формы документации можно охарактеризовать следующим образом:

- ▣ **Требования** определяют, для чего нужна программа;
- ▣ **Цели** устанавливают, что именно и насколько хорошо должна делать программа;
- ▣ **Внешние спецификации** дают точное описание способа представления программы пользователям;
- ▣ **Документация**, связанная с последующими процессами, определяет способ построения программы на различных уровнях детализации.

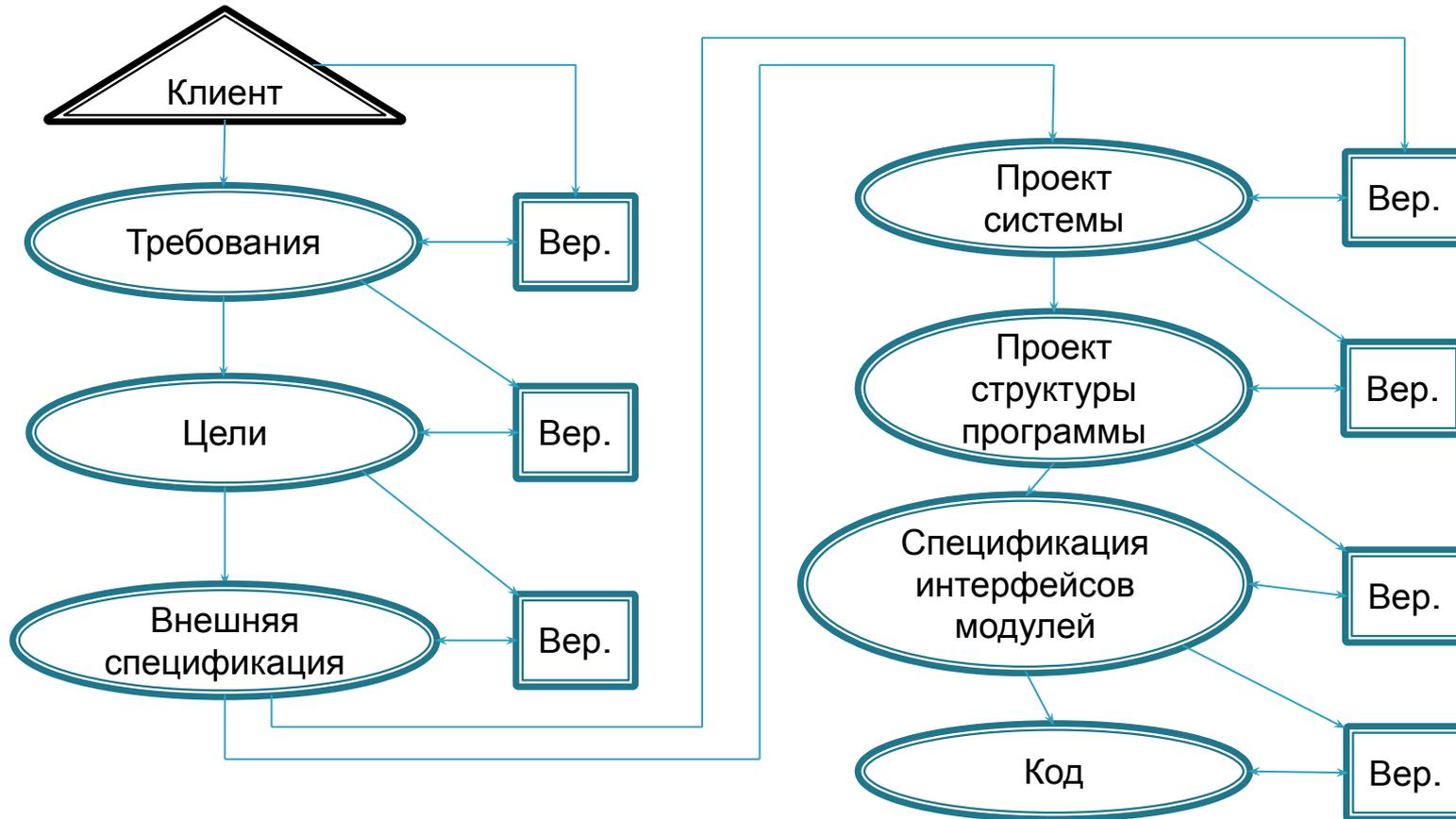
Процесс разработки ПО

Именно в ходе выполнения данных семи этапов(в ходе передачи, осмысления и преобразования информации) появляются большинство ошибок.

Три подхода к уменьшению количества ошибок на данных этапах:

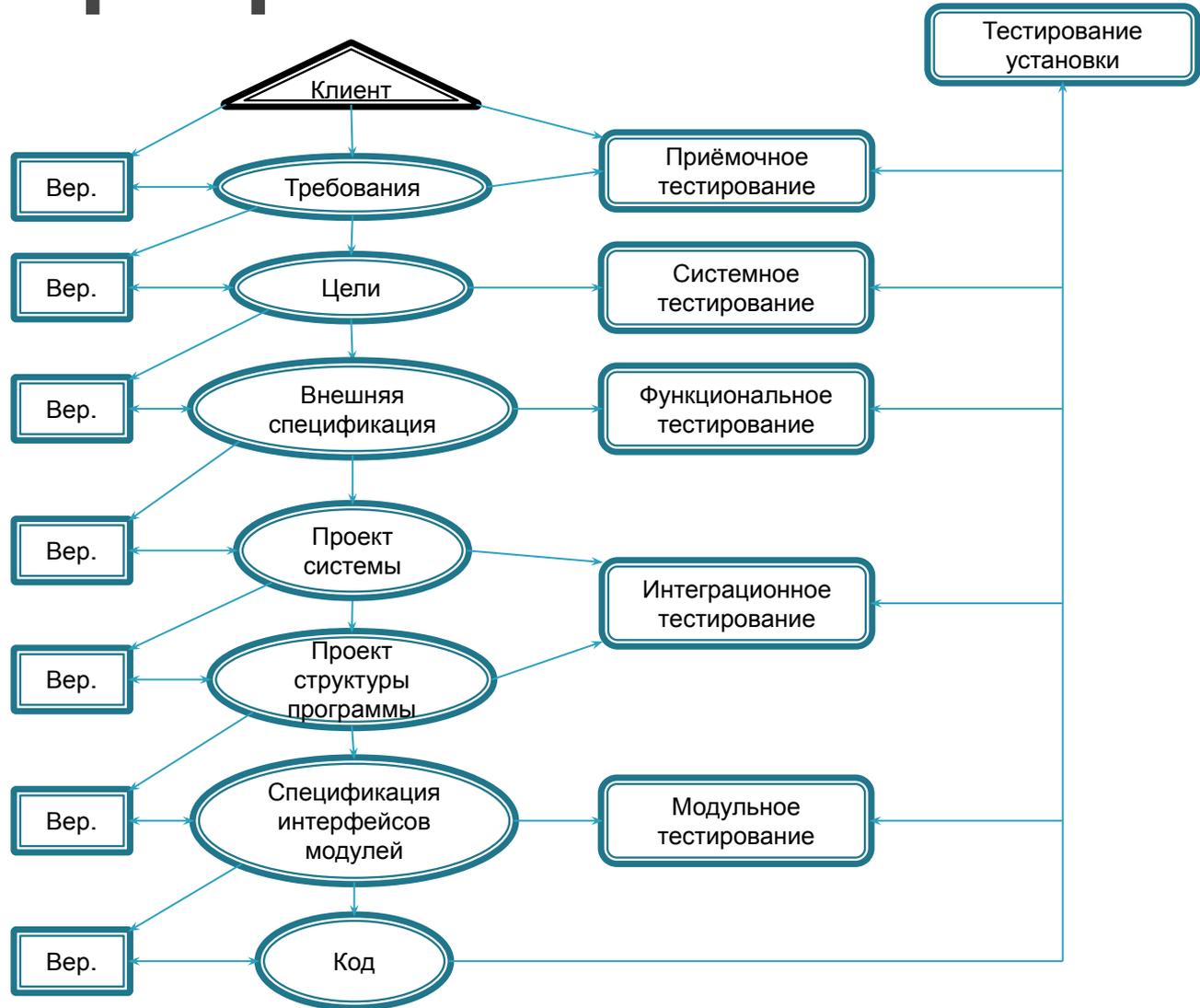
- 1) Повысить качество разработки каждого этапа;

Процесс разработки ПО



2) Введение после каждого этапа разработки, отдельной стадии верификации результатов

Процесс разработки ПО



Процесс разработки ПО

3) Суть подхода - ориентировать конкретные процессы тестирования на конкретные процессы разработки, т.е. фокусировать каждый процесс тестирования на вполне определенном этапе преобразования информации в другую форму и, таким образом, на определённом классе ошибок.

Цикл тестирования структурно встраивается в модель цикла разработки, т.е. на каждом этапе возможно установить взаимно однозначное соответствие между процессами разработки и тестирования. Например:

- Цель модульного теста – найти расхождения между характеристиками модулей программы и спецификации их интерфейсов;
- Цель функционального теста – продемонстрировать, что программа не соответствует внешним спецификациям;

Процесс разработки ПО

Обратите внимание на то, что все тесты направлены на поиск и локализацию ошибок. Т.е. мы изначально уверены, что ошибки присутствуют.

В случае, если тестировщик уверен, что ошибок нет, то тестирование будет изначально не полным.

Преимущества 3-го подхода в том, что он позволяет избежать избыточного ненужного тестирования и в то же время исключает возможность пропуска большого числа классов ошибок.

Недостатки такого подхода – использование возможно лишь на крупных проектах, для которых присутствует контрактная документация, ТЗ, возможно широкое применение.

Для небольших программ зачастую достаточно функционального тестирования.

Функциональное тестирование

Функциональное тестирование – процесс направленный на выявление расхождений между поведением программы и внешней спецификацией.

Внешняя спецификация – это точное описание поведения программы с точки зрения конечного пользователя.

- Выполняется в рамках стратегии «черного ящика»;
- Ожидается, что выполнено модульное тестирование;
- Выполнению предшествует анализ спецификации с целью составления наборов тестов;

Системное тестирование

Системное тестирование – сопоставляет результат реализации системы или программы с первоначально сформулированными для неё целями.

- Системное тестирование не ограничивается только системами. Если продукт является программой, то системное тестирование в данном случае представляет собой попытку продемонстрировать, какие из стоящих перед программой целей не реализуются;
- Системное тестирование по определению не возможно, если отсутствует документ, отражающий набор измеримых целей, достижение которых возлагается на продукт;

Системное тестирование

- Основные ошибки – на этапе трансляции, допущенные в процессе проектирования внешней спецификации;
- Нельзя использовать внешнюю спецификацию для основания системного тестирования, поскольку это противоречит цели системного тестирования;
- Документа, описывающего целевые задачи системы, поскольку такой документ не может содержать точных описаний внешних интерфейсов;

Системные тесты проектируются на основе документированных целей программы, а создаются на основании пользовательской документации к проекту.

Системное тестирование

Категория	Описание
Возможности (Facility)	Проверяется полнота реализации функциональных возможностей, определённых целями
Предельные объемы данных (Volume)	Проверяется способность программы обрабатывать аномально большие объемы данных
Нагрузочное тестирование (Stress)	Проверяется работоспособность программы при повышенных нагрузках, обычно в условиях параллельной обработки данных
Удобство использования (Usability)	Определяется, насколько удобно конечному пользователю взаимодействовать с программой
Безопасность (Security)	Предпринимаются попытки обойти средства защиты программы
Производительность (Performance)	Определяется соответствие программы требованиям производительности и скорости отклика
Память (Storage)	Проверяется способность системы эффективно использовать как оперативную, так и долговременную память

Системное тестирование

Категория	Описание
Конфигурация (Configuration)	Проверяется работоспособность системы в рекомендованных конфигурациях
Совместимость (Compatibility)	Определяется совместимость новых версий программы с предыдущими версиями
Установка (Installation)	Проверяется работоспособность методов установки программы на всех поддерживаемых платформах
Надежность (Reliability)	Определяется соответствие программы специфицированным показателям надежности, таким как длительность непрерывной работы и среднее время наработки на отказ
Восстанавливаемость (Recovery)	Тестируется способность средств восстановления системы выполнять свои функции
Обслуживаемость (Serviceability)	Определяется, обеспечивает ли приложение механизмы предоставления данных о событиях, требующих оказания технической поддержки
Документированность (Documentation)	Проверяется точность всей пользовательской документации

Системное тестирование

Категория	Описание
Процедуры (Procedure)	Определяется точность специальных процедур, которые должны соблюдаться в процессе обслуживания или использования программы

Интеграционное тестирование

Интеграционное тестирование предназначено для проверки связи между компонентами, а также взаимодействия с различными частями системы (операционной системой, оборудованием либо связи между различными системами).

Два уровня:

- 1) Компонентный интеграционный уровень. Проверяется взаимодействие между компонентами системы после проведения модульного тестирования.
- 2) Системный интеграционный уровень. Проверяется взаимодействие между разными системами после проведения системного тестирования.

Интеграционное тестирование

Подходы:

1) Снизу-вверх. Все низкоуровневые модули, процедуры или функции собираются воедино и затем тестируются. После чего собирается следующий уровень модулей для проведения интеграционного тестирования. Данный подход считается полезным, если все или практически все модули, разрабатываемого уровня, готовы. Также данный подход помогает определить по результатам тестирования уровень готовности приложения.

2) Сверху вниз. Вначале тестируются все высокоуровневые модули, и постепенно один за другим добавляются низкоуровневые. Все модули более низкого уровня симулируются заглушками с аналогичной функциональностью, затем по мере готовности они заменяются реальными активными компонентами. Таким образом мы проводим тестирование сверху вниз.

Интеграционное тестирование

3) Большой взрыв. Все или практически все разработанные модули собираются вместе в виде законченной системы или ее основной части, и затем проводится интеграционное тестирование. Такой подход очень хорош для сохранения времени. Однако если тест кейсы и их результаты записаны не верно, то сам процесс интеграции сильно осложнится, что станет преградой для команды тестирования при достижении основной цели интеграционного тестирования.

Тестирование установки

Тестирование установки – это процесс, цель которого выявить ошибки в процессе установки программы.

- Не входит в этап разработки ПО;
- Опытный разработчик всегда сам выполняет предварительное приёмочное тестирование.