

# SimbirSoft

## Уровни, типы, виды, методы, подходы тестирования

Ведин Алексей  
QA-Специалист

07.06.2018

# Уровни тестирования



# Уровни тестирования.

## Компонентное тестирование

**Компонентное тестирование** (Component or Unit Testing) - это тестирование программы на уровне отдельно взятых модулей, функций или классов.

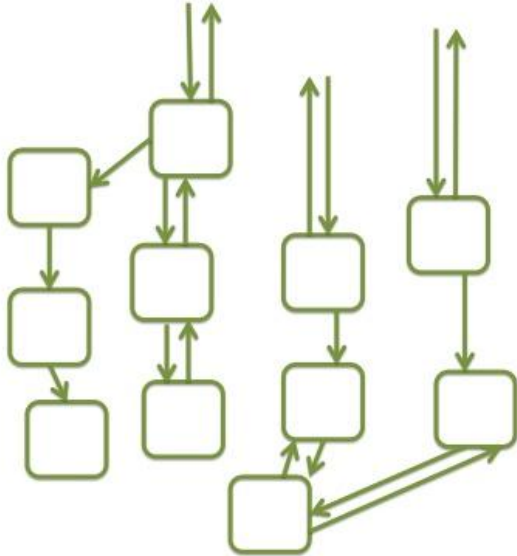
**Цель:** выявление локализованных в модуле ошибок в реализации алгоритмов, а также определение степени готовности системы к переходу на следующий уровень разработки и тестирования.

**Unit-тесты** - тесты проверяющие корректность работы отдельных модулей программы.



# Уровни тестирования.

## Интеграционное тестирование



**Интеграционные тесты** – тесты, проверяющие корректность взаимодействия отдельных модулей друг с другом.



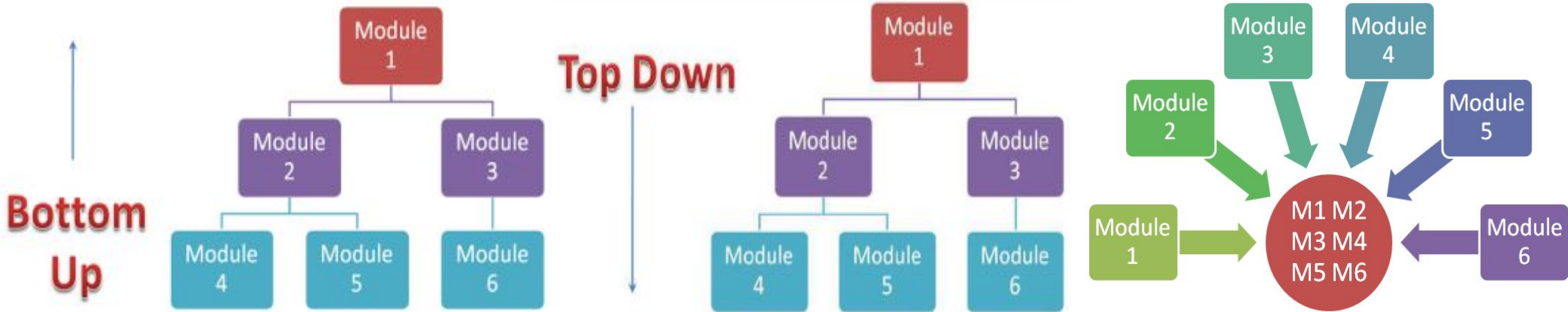
26

**Компонентный интеграционный уровень (*Component Integration testing*)** - проверяется взаимодействие между компонентами системы после проведения компонентного тестирования.

**Системный интеграционный уровень (*System Integration Testing*)** - проверяется взаимодействие между разными системами после проведения системного тестирования.

# Уровни тестирования.

## Интеграционное тестирование. Подходы



### Снизу вверх (*Bottom Up Integration*)

Все низкоуровневые модули, процедуры или функции собираются воедино и затем тестируются. После чего собирается следующий уровень модулей для проведения интеграционного тестирования. Считается полезным, если все или практически все модули, разрабатываемого уровня, готовы.

### Сверху вниз (*Top Down Integration*)

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.

### Большой взрыв (*"Big Bang" Integration*)

Все разработанные модули собираются вместе в виде законченной системы или ее основной части, и затем проводится интеграционное тестирование. Подход *"Big Bang"* очень хорош для экономии времени. Но в тоже время при таком подходе довольно тяжело локализовать ошибки.

# Уровни тестирования.

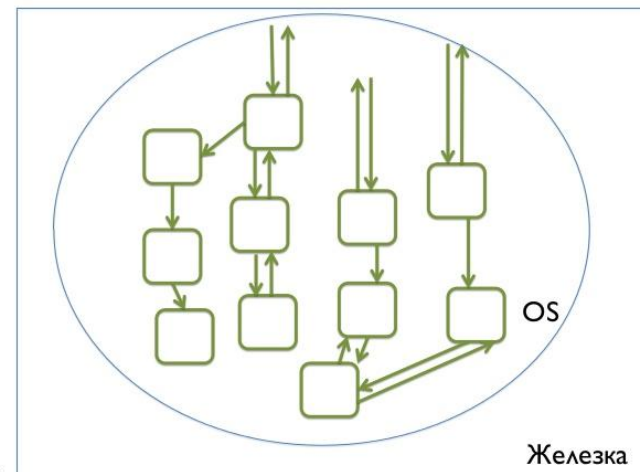
## Системное тестирование

Основной задачей системного тестирования является проверка как функциональных, так и не функциональных требований в системе в **целом**.

При этом выявляются дефекты, такие как неверное использование ресурсов системы, непредусмотренные комбинации данных пользовательского уровня, несовместимость с окружением, непредусмотренные сценарии использования, отсутствующая или неверная функциональность, неудобство использования и т.д.

Для минимизации рисков, связанных с особенностями поведения системы в той или иной среде, во время тестирования **следует использовать окружение максимально приближенное к тому, на которое будет установлен продукт.**

Даже если все компоненты по отдельности работают правильно и корректно взаимодействуют друг с другом – это ещё ничего не значит.



# Уровни тестирования. Системное тестирование. Подходы

## На базе требований (*requirements based*)

Форма содержит поля:

- Пользователь
- Полное имя
- Описание
- Пароль/подтверждение пароля

A screenshot of a software window titled "Новый пользователь" (New User). The form contains the following fields and options:

- Пользователь: User99
- Полное имя: (empty)
- Описание: Временный сотрудник
- Пароль: (masked with dots)
- Подтверждение: (masked with dots)
- Options:
  - Потребовать смену пароля при следующем входе в систему
  - Запретить смену пароля пользователям
  - Срок действия пароля не ограничен
  - Отключить учетную запись
- Buttons: Создать, Закрыть

## На базе случаев использования (*use case based*)

Начало работы в системе:

Администратор заводит новых пользователей, указав необходимые данные. Система делает проверку на уникальность логина и пароля и создает запись. Если логин не уникален, система выдает сообщение и предлагает либо изменить логин, либо отменить создание пользователя.



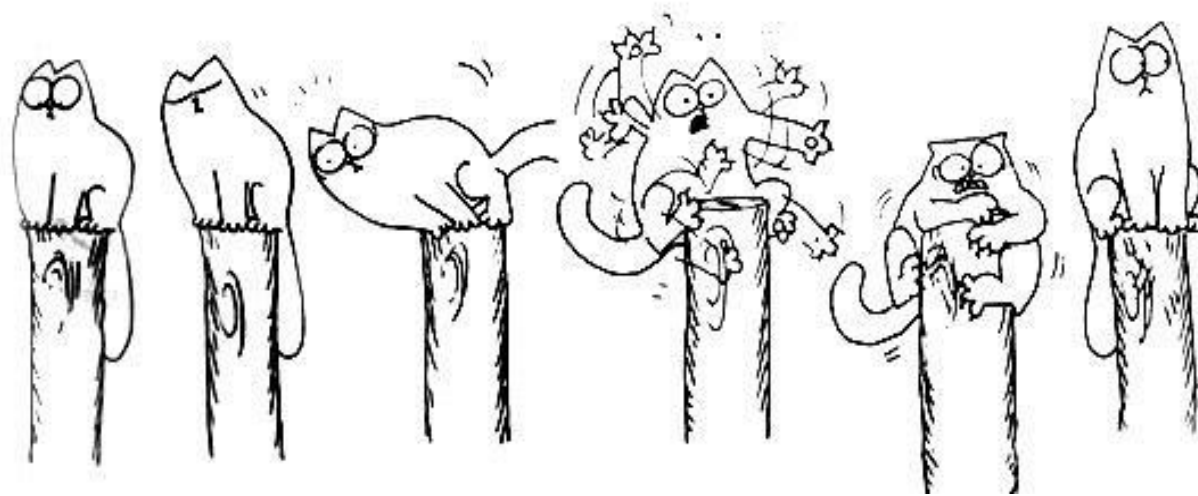
# Уровни тестирования.

## Приемочное тестирование

Формальный процесс тестирования, который проверяет работоспособность системы, частей системы или отдельных нефункциональных характеристик системы и проводится с целью: определения, удовлетворяет ли система приемочным критериям; вынесения решения заказчиком или другим уполномоченным лицом, принимается приложение или нет.

**Поиск дефектов не является главной целью приемочного тестирования.**

**Фаза приемочного тестирования** длится до тех пор, пока заказчик не выносит решение об отправлении приложения на доработку или выдаче приложения.





# Уровни тестирования.

## Приемочное тестирование

**Пользовательское** - проверяет готовность системы для использования в бизнесе.



**Эксплуатационное** - приемочное тестирование, проводимое системными администраторами (тестирование резервного копирования\восстановления, восстановление после сбоев, управление пользователями и пр.)



**Контрактное и правовое** - выполняется для проверки требований, предъявляемых контрактом к разрабатываемому ПО.



**Альфа и бета тестирование (или тестирование в условиях эксплуатации):**

**Альфа-тестирование** выполняется организацией, разрабатывающей продукт, но не группой разработчиков.

**Бета-тестирование**, выполняется покупателями или потенциальными заказчиками на их собственных мощностях.



# Типы тестирования

Функциональное  
тестирование

Нефункциональное  
тестирование

Структурное  
тестирование

Тестирование  
изменений

# Типы тестирования.

## Функциональное тестирование

Функциональные тесты базируются на функциях, а также взаимодействии с другими системами, и могут быть представлены на всех уровнях тестирования: компонентном, интеграционном, системном и приемочном. Функциональные виды тестирования рассматривают внешнее поведение системы, т.е. **функции, которые описывают, «что» система делает.**

**Самые распространенные виды функциональных тестов:**

Функциональное тестирование (Functional testing)

Тестирование взаимодействия (Interoperability Testing)

Тестирование безопасности (Security and Access Control Testing)

# Типы тестирования.

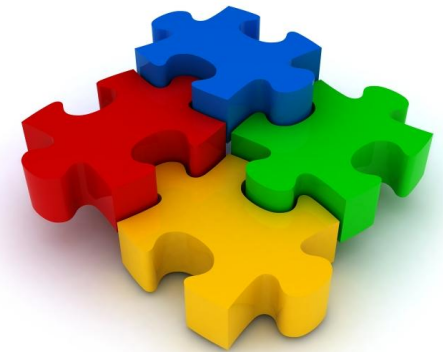
## Функциональное тестирование. Виды

Функциональное тестирование [\(Functional testing\)](#)

основывается на знании о поведении системы, которое описывается в проектной документации.

**Тестирование ВЗАИМОДЕЙСТВИЯ (Interoperability Testing)**

- функциональное тестирование, проверяющее способность приложения взаимодействовать с одним и более компонентами или системами, и включающее в себя тестирование **совместимости** (compatibility testing) и **интеграционное тестирование** ([integration testing](#)).



# Типы тестирования.

## Функциональное тестирование. Виды

**Тестирование безопасности** - вид тестирования, используемый для проверки безопасности системы, для анализа рисков, связанных с обеспечением защиты приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным.



### Принципы безопасности программного обеспечения:

1. **Конфиденциальность** (сокрытие определенных ресурсов или информации).
2. **Целостность** (критерии: доверие, повреждение и восстановление).
3. **Доступность** (представляет собой требования о том, что ресурсы должны быть доступны авторизованному пользователю, внутреннему объекту или устройству, как правило, чем более критичен ресурс, тем выше уровень доступности должен быть).

# Типы тестирования.

## Нефункциональное тестирование

Тестирование того, «как» система работает.

Самые распространенные виды нефункциональных тестов:

Тестирование пользовательского интерфейса (UI/GUI)

Тестирование удобства пользования (Usability testing)

Тестирование локализации (localization testing)

Конфигурационное тестирование (configuration testing)

Тестирование производительности (Performance and Load testing)

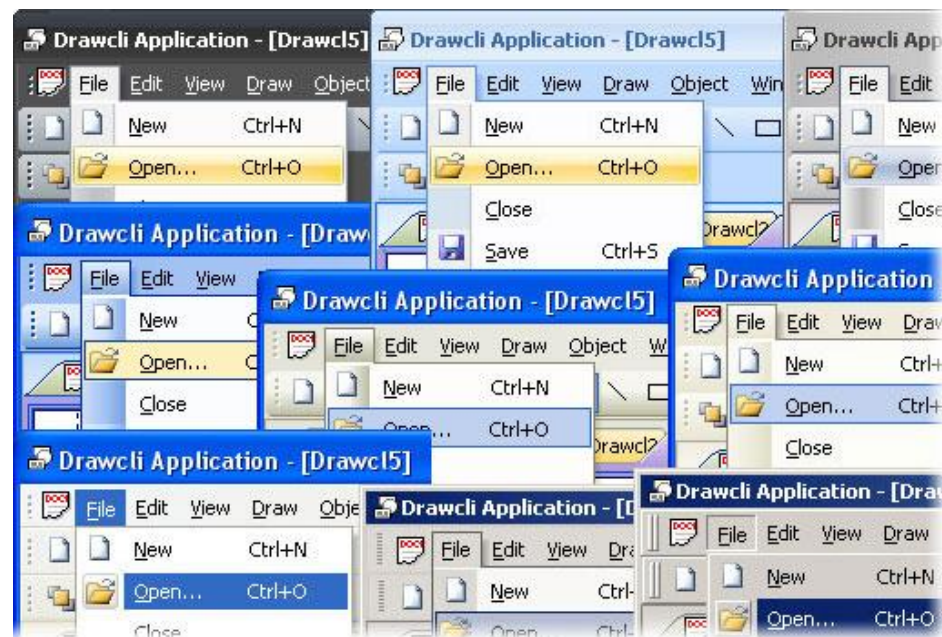
Тестирование инсталляции (Installation testing)

# Типы тестирования.

## Нефункциональное тестирование. Виды

### Тестирование пользовательского интерфейса (GUI/UI)

- Тестирование с различными разрешениями экрана;
- Тестирование на соответствие стандартам графических интерфейсов;
- Тестирование в ограниченных условиях, например, в условиях нехватки памяти;
- Тестирование графического интерфейса пользователя на целевых устройствах;
- Тестирование локализованных версий: точность перевода, проверка длины названий элементов интерфейса и т.д.;



# Типы тестирования.

## Нефункциональное тестирование. Виды

**Тестирование удобства пользования (usability testing)** - это вид тестирования, направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта.



Приложение оценивается по следующим пунктам:

**производительность, эффективность (efficiency)** - сколько времени и шагов понадобится пользователю для завершения основных задач приложения, например, размещение новости, регистрации, покупка и т.д.? *(меньше - лучше)*

**правильность (accuracy)** - сколько ошибок сделал пользователь во время работы с приложением? *(меньше - лучше)*

**активизация в памяти (recall)** – как много пользователь помнит о работе приложения после приостановки работы с ним на длительный период времени? *(повторное выполнение операций после перерыва должно проходить быстрее чем у нового пользователя)*

**эмоциональная реакция (emotional response)** – как пользователь себя чувствует после завершения задачи - растерян, испытал стресс? Порекомендует ли пользователь систему своим друзьям? *(положительная реакция - лучше)*



# Типы тестирования.

## Нефункциональное тестирование. Виды

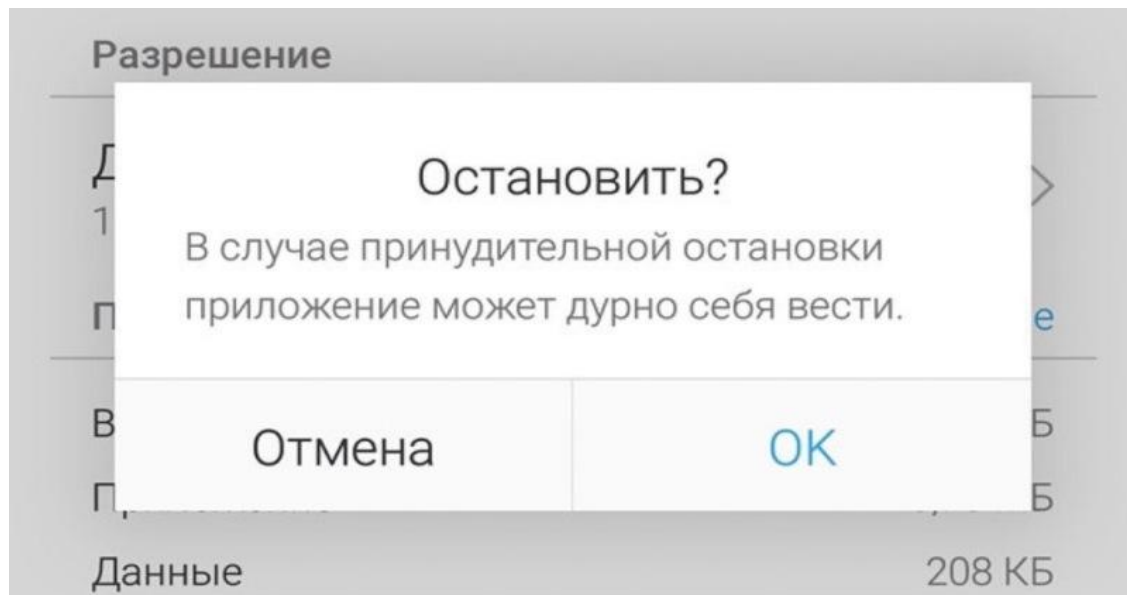


**Тестирование локализации (localization testing)** - это процесс тестирования локализованной версии программного продукта.

**Цель теста локализации** – убедиться, что приложение поддерживает многоязыковый интерфейс и функции.

**Необходимо обращать внимание на:**

- перевод на другой язык,
- формат дат и чисел,
- порядок имени и фамилии,
- валюты и т.д.



# Типы тестирования.

## Нефункциональное тестирование. Виды

### Конфигурационное тестирование или тестирование портируемости

(*Configuration Testing* или *Portability testing*) — специальный вид тестирования, направленный на проверку взаимодействия программного обеспечения с окружением, в которое оно будет установлено (Аппаратные и Программные средства), а так же с позиции конечного пользователя и конфигурации его рабочей станции.

### Можно выделить 2 цели конфигурационного тестирования:

- Определить оптимальную конфигурацию оборудования, обеспечивающую требуемые характеристики производительности и времени реакции тестируемой системы.
- Проверить объект тестирования на совместимость с объявленным в спецификации оборудованием, операционными системами и программными продуктами третьих фирм.



**Configuration Testing**

# Типы тестирования.

## Нефункциональное тестирование. Виды

Уровни проведения конфигурационного тестирования:

- Серверный
- Клиентский

На серверном уровне тестируется взаимодействие выпускаемого программного обеспечения с окружением, в которое оно будет установлено:

- Аппаратные средства (тип и количество процессоров, объем памяти, характеристики сети / сетевых адаптеров и т.д.)
- Программные средства (ОС, драйвера и библиотеки, стороннее ПО, влияющее на работу приложения и т.д.)

Основной упор здесь делается на тестирование с целью определения оптимальной конфигурации оборудования, удовлетворяющего требуемым характеристикам качества (эффективность, портативность, удобство сопровождения, надежность).



# Типы тестирования.

## Нефункциональное тестирование. Виды

Уровни проведения конфигурационного тестирования:

- Серверный
- Клиентский



**На клиентском уровне** программное обеспечение тестируется с позиции конечного пользователя и конфигурации его рабочей станции:

1. Тип, версия и битность операционной системы (подобный вид тестирования называется **кросс-платформенное тестирование**)
2. Тип и версия Web-браузера, в случае если тестируется Web-приложение (подобный вид тестирования называется **кросс-браузерное тестирование**)
3. Тестирование работы при различных версиях драйверов
4. Тип и модель видеоадаптера (при тестировании игр это очень важно)

и т.д.

# Типы тестирования.

## Нефункциональное тестирование. Виды

**Тестирование производительности (Performance and Load Testing)** - это автоматизированное тестирование, имитирующее работу определенного количества пользователей на каком-либо общем (разделяемом ими) ресурсе.



**Нагрузочное тестирование (Load testing)** - проводится для того, чтобы оценить поведение приложения под заданной ожидаемой нагрузкой.



**Объемное тестирование (Volume Testing)** - для получения оценки производительности при увеличении объемов данных в базе данных приложения.



**Стрессовое тестирование (Stress Testing)** - насколько приложение работоспособно в условиях стресса и также оценить способность системы к регенерации, т.е. к возвращению к нормальному состоянию после прекращения воздействия стресса.



**Тестирование стабильности (Stability/Reliability Testing)** - проверка работоспособности приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки.

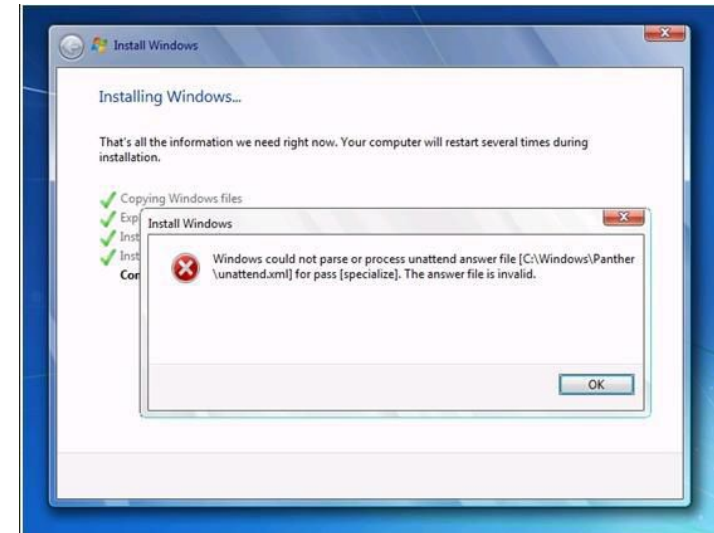
# Типы тестирования.

## Нефункциональное тестирование

### Тестирование инсталляции (Installation testing)

Тестирование процедуры инсталляции продукта разделяется на несколько частей:

- Формальное тестирование программы установки (графический интерфейс пользователя, общее удобство пользования, соответствие стандартам);
- Функциональное тестирование программы установки;
- Тестирование механизма лицензирования и способности противостоять взлому;
- Проверка успешной инсталляции и настройки;
- Проверка стабильности работы приложения после установки;
- Тестирование обновления приложения;
- Тест удаления программного обеспечения.



# Типы тестирования.

## Структурное тестирование

**Структурное тестирование** основывается на детальном изучении логики программы и подборе тестов, обеспечивающих максимально возможное число проверяемых операторов, логических ветвлений и условий.



**Структурное тестирование может выполняться на всех уровнях тестирования.**

Структурные методы тестирования лучше всего использовать после методов разработки тестов на основе спецификации, чтобы измерить тщательность тестирования, используя измерения покрытия структуры программы.

# Типы тестирования.

## Тестирование изменений

После проведения необходимых изменений, таких как исправление дефекта, разработка новой функциональности, программное обеспечение должно быть перетестировано для подтверждения того факта, что проблема была действительно решена или новая функциональность не привела к новым дефектам или полному отказу системы.

**Виды тестирования, которые необходимо проводить для подтверждения работоспособности приложения или правильности исправления дефекта:**

Дымовое тестирование  
(Smoke Testing)

Тестирование сборки (Build  
Verification Test)

Повторное (подтверждающее)  
тестирование (Re-testing)

Регрессионное тестирование  
(Regression Testing)



# Типы тестирования.

## Тестирование изменений. Виды

**Дымовое тестирование или Smoke Testing** - короткий цикл тестов, выполняемый для подтверждения того, что после сборки кода (нового или исправленного) устанавливаемое приложение стартует и выполняет основные функции.



**Тестирование сборки (Build verification testing)** - тестирование, направленное на определение соответствия выпущенной версии критериям качества для начала тестирования.

# Типы тестирования.

## Тестирование изменений. Виды

**Повторное (подтверждающее) тестирование (Re-testing)** - после того, как дефект обнаружен и исправлен, программу необходимо перепроверить, чтобы убедиться, что исходный дефект успешно устранен. Это называется подтверждением.



99 маленьких багов в коде,  
99 маленьких багов в коде,  
Один нашли, пофиксили.

127 маленьких багов в коде



**Регрессионное тестирование (Regression Testing)** - направлены на проверку изменений, сделанных в приложении или окружающей среде (починка дефекта, слияние кода, миграция на другую операционную систему, базу данных, веб-сервер или сервер приложения), для подтверждения того факта, что существующая ранее функциональность работает как и прежде.

# Методы тестирования

Статический  
метод

Динамический  
метод

# Методы тестирования.

## Статический метод

**Рецензирование** - вид тестирования ПО (включая код), который может проводиться перед динамическим тестированием. Осуществляется без запуска приложения.



Может проводится для любого продукта, связанного с разработкой ПО: спецификации требований и дизайна, код, планы тестирования, руководства пользования и т.п. Во время рецензирования могут быть найдены упущения, например, в требованиях, которые маловероятно найти во время динамического тестирования.

**Статический анализ** - это исследование ПО с помощью специального инструмента без его запуска. Инструментальные средства статического анализа анализируют: код программы, сгенерированный код, например HTML, XML.

Типичные дефекты, которые могут быть найдены при статическом анализе: переменные, которые не используются или некорректно объявлены, невыполняемые ветки кода, пропущенная или неверная логика (например, бесконечные циклы), отклонение от стандартов программирования, нарушение синтаксиса и тд

# Методы тестирования. Динамический метод.

Динамические методы тестирования подразумевают запуск программы.

## Методы черного ящика (black box testing)

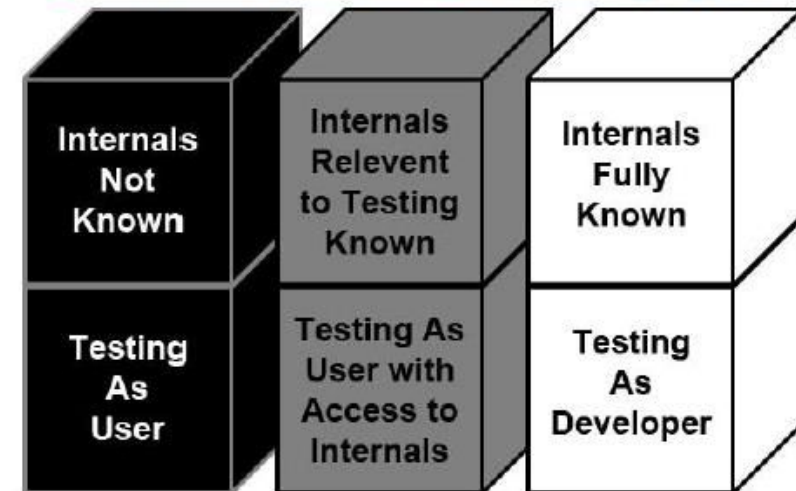
**Функциональное тестирование.** Тестирование на соответствие программного продукта требованиям без знания внутренней структуры реализации системы.

## Методы белого ящика (white/glass box testing)

**Структурное тестирование.** Тестирование на соответствие программного продукта требованиям со знанием внутренней структуры реализации системы (есть в наличии исходный код и технические спецификации). Позволяет проводить локализацию ошибок, анализ надежности и устойчивости, существенно повышая качество системы.

**Методы серого ящика (grey box testing)** Комбинирование методов белого и черного ящика. При тестировании серого ящика разработчик теста имеет доступ к исходному коду, но при непосредственном выполнении тестов доступ к коду, как правило, не требуется.

Differences Between Box Testing Types



# Подходы к тестированию

Сценарное  
тестирование

Исследовательское  
тестирование  
(exploratory testing)

Интуитивное  
тестирование  
(Ad-hoc testing)

# Подходы к тестированию.

## Сценарное тестирование

**Сценарное тестирование (scenario testing):** есть документация (функциональные спецификации, спецификации требований, руководства пользователя и т.д.), первоначально осуществляется анализ и проектирование тестовых сценариев и тестовых случаев на основании документации, тестирование выполняется по созданным тест-кейсам.

**Исследовательское тестирование (exploratory testing):** нет документации, проектирование и выполнение тест-кейсов происходит одновременно, по мере тестирования составляется пользовательский сценарий.

**Ad-hoc тестирование (Ad-hoc testing):** нет документации, нет тест-кейсов и они не проектируются, пользователь о приложении не знает ничего, негативное тестирование, тесты придумываются на лету.



# Вопросы?





## Домашнее задание

1. Описать какие Уровни, Типы, Виды, Методы, Подходы к тестированию Вы бы использовали при тестировании приложения

<http://fcs.simbirsoft1.com/>.

2. По каждой классификации тестирования описываем свои мысли.

Например: Безопасность: Сессия должна иметь ограниченное время жизни из соображений безопасности, а в приложении бесконечное время жизни сессии, пока не закроешь браузер.

3. Если по какой-либо из классификаций тестирования проверка невозможна, указываем это.

Например: Тестирование инсталляции не проводится, так как это web-приложение.

Цель домашнего задания - научиться использовать классификацию тестирования при процессе тестирования, а не найти баги. Формат отчета может быть выбран на Ваше усмотрение.



+7 (8422) 44-66-91

+7 (495) 133-90-01

[www.simbirsoft.com](http://www.simbirsoft.com)

# Кратко

## Уровни тестирования:

- ❖ Компонентное тестирование (Component or Unit Testing)

- ❖ Интеграционное тестирование (Integration testing)

Компонентный интеграционный уровень (Component Integration testing)

Системный интеграционный уровень (System Integration Testing)

- ❖ Системное тестирование (System Testing)

На базе требований (requirements based)

На базе случаев использования (use case based)

- ❖ Приёмочное тестирование (Acceptance Testing)

Пользовательское (User Acceptance Testing)

Эксплуатационное (Operational testing)

Контрактное и правовое (Contract/Legal/Law testing)

Альфа тестирование (alpha testing)

Бета тестирование (beta testing)

# Кратко

## Типы и виды тестирования:

### ❖ Функциональное (Functional testing)

Функциональное (Functional testing)

Тестирование взаимодействия (Interoperability Testing), включает в себя:  
совместимости (compatibility testing) и интеграционное тестирование (integration testing)

Тестирование безопасности (Security and Access Control Testing)

### ❖ Нефункциональное (Non-functional testing)

Тестирование пользовательского интерфейса (UI/GUI)

Тестирование удобства пользования (Usability testing)

Тестирование локализации (localization testing)

Конфигурационное тестирование (configuration testing), включает в себя:  
кросс-платформенное тестирование и кросс-браузерное тестирование

Тестирование производительности (Performance and Load testing), включает в себя:  
нагрузочное (Load testing), объемное (Volume Testing), стрессовое (Stress Testing), стабильности (Stability/Reliability Testing)

Тестирование инсталляции (Installation testing)

### ❖ Структурное (structural testing)

### ❖ Изменений (changes testing)

Дымовое тестирование (Smoke Testing)

Тестирование сборки (Build Verification Test)

Повторное (подтверждающее) тестирование (Re-testing)

Регрессионное тестирование (Regression Testing)

# Кратко

## Методы тестирования:

- ❖ Статические методы (Static methods)

Рецензирование (Reviewing)

Статический анализ (Static analysis)

- ❖ Динамические методы (Dynamic methods)

Методы черного ящика (black box testing)

Методы белого ящика (white/glass box testing)

Методы серого ящика (grey box testing)

## Подходы к тестированию:

- ❖ Сценарное тестирование (Scenario testing)
- ❖ Исследовательское тестирование (exploratory testing)
- ❖ Интуитивное тестирование (Ad-hoc testing)

Функциональное (Functional testing)

Тестирование взаимодействия (Interoperability Testing)

Тестирование безопасности (Security and Access Control Testing)

Тестирование пользовательского интерфейса (UI/GUI)

Тестирование удобства пользования (Usability testing)

Тестирование локализации (localization testing)

Конфигурационное тестирование (configuration testing),

Тестирование производительности (Performance and Load testing),

Тестирование инсталляции (Installation testing)

Написать 5 проверок на тестирование ластика,  
используя вид тестирования ваш вид  
тестирования



# Спасибо за внимание!

Ведин Алексей  
QA-Специалист

+7 (8422) 44-66-91

+7 (495) 133-90-01

[www.simbirsoft.com](http://www.simbirsoft.com)