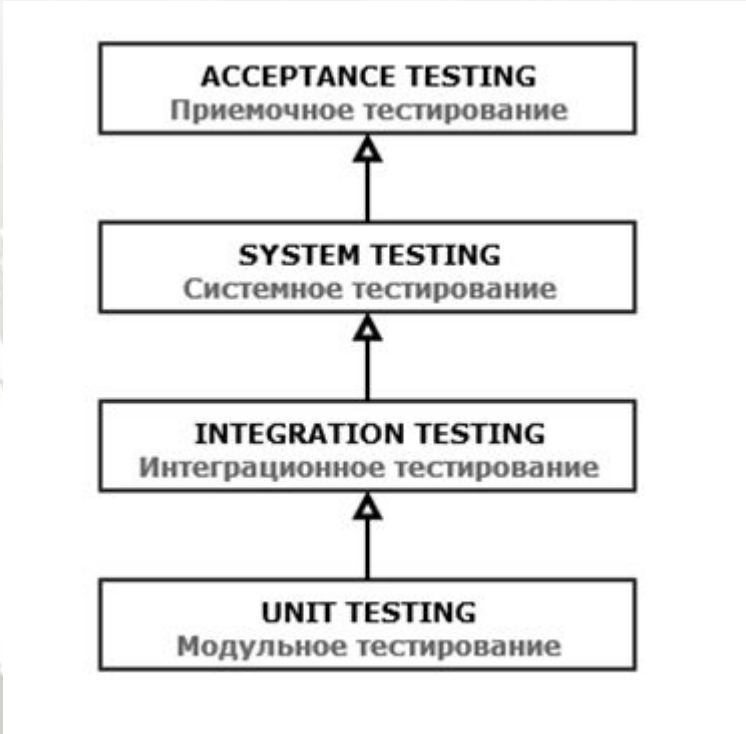
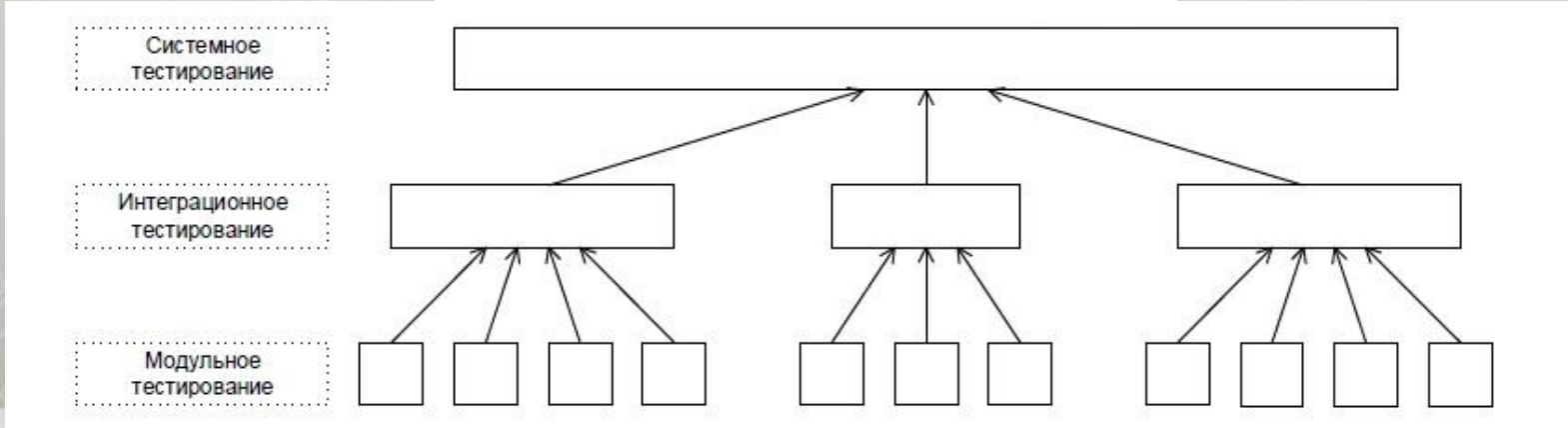
Testing levels and classifications
Уровни и виды тестирования

Test Levels

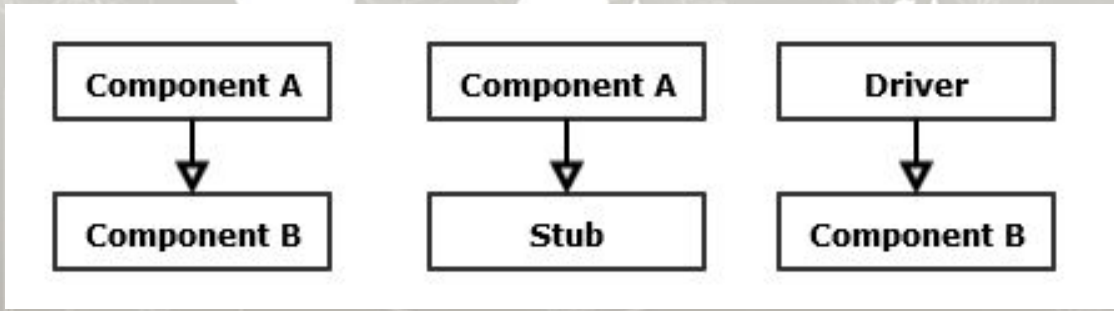

ACCEPTANCE TESTING
Приемочное тестирование
SYSTEM TESTING
Системное тестирование
INTEGRATION TESTING
Интеграционное тестирование
UNIT TESTING
Модульное тестирование


Системное тестирование
Интеграционное тестирование
Модульное тестирование

Test Levels

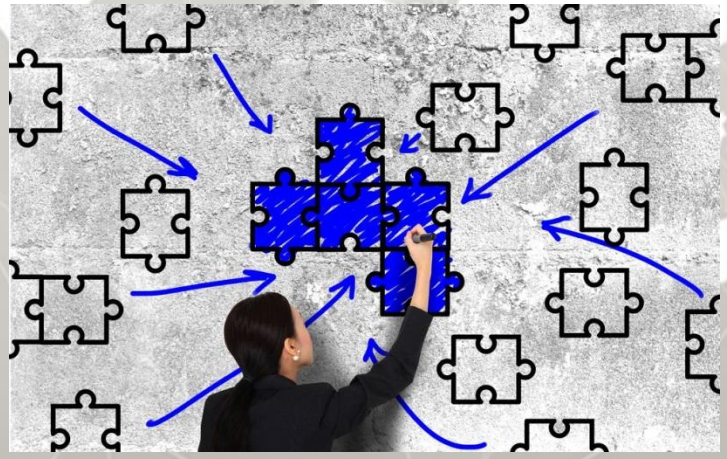
Модульное (компонентное) тестирование (*unit testing, module testing, component testing*) направлено на проверку отдельных небольших частей приложения, которые (как правило) можно исследовать изолированно от других подобных частей.

«Заглушка» вызывается компонентой, которая тестируется, а драйвер дергает компоненту, которая должна быть протестирована.



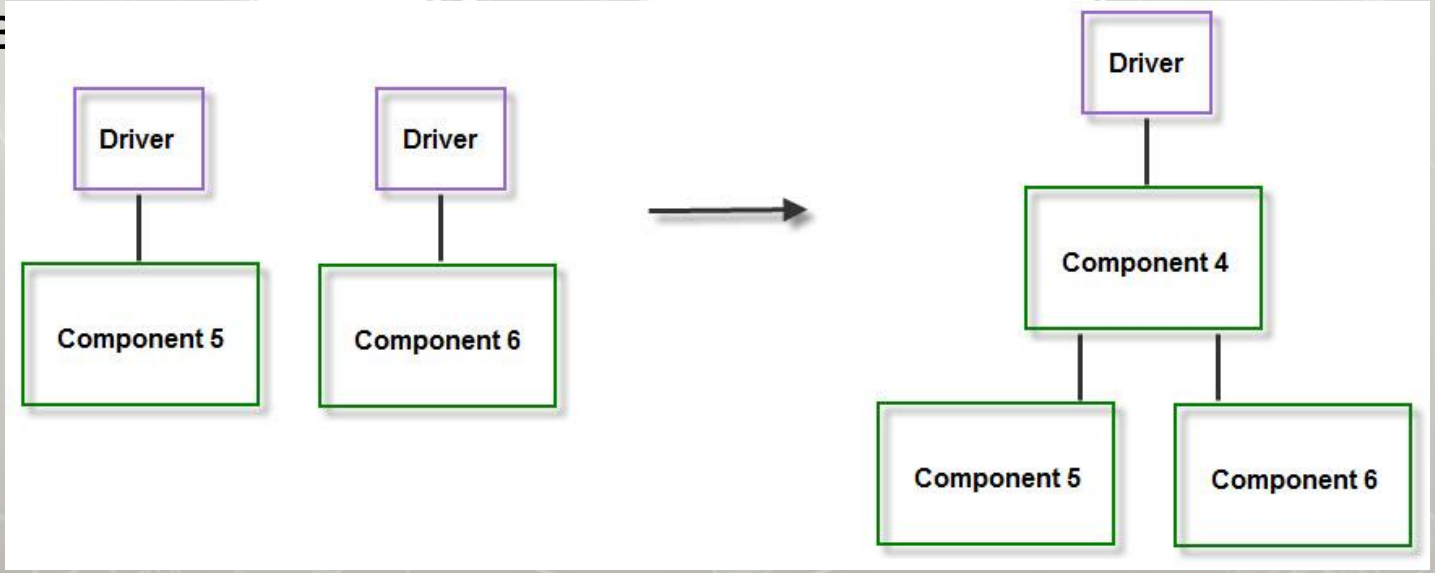
Test Levels

Интеграционное тестирование (*integration testing*) направлено на проверку взаимодействия между несколькими частями приложения (каждая из которых, в свою очередь, проверена отдельно на стадии модульного тестирования), а также взаимодействия с различными частями системы (операционной системой, оборудованием либо связи между различными системами).



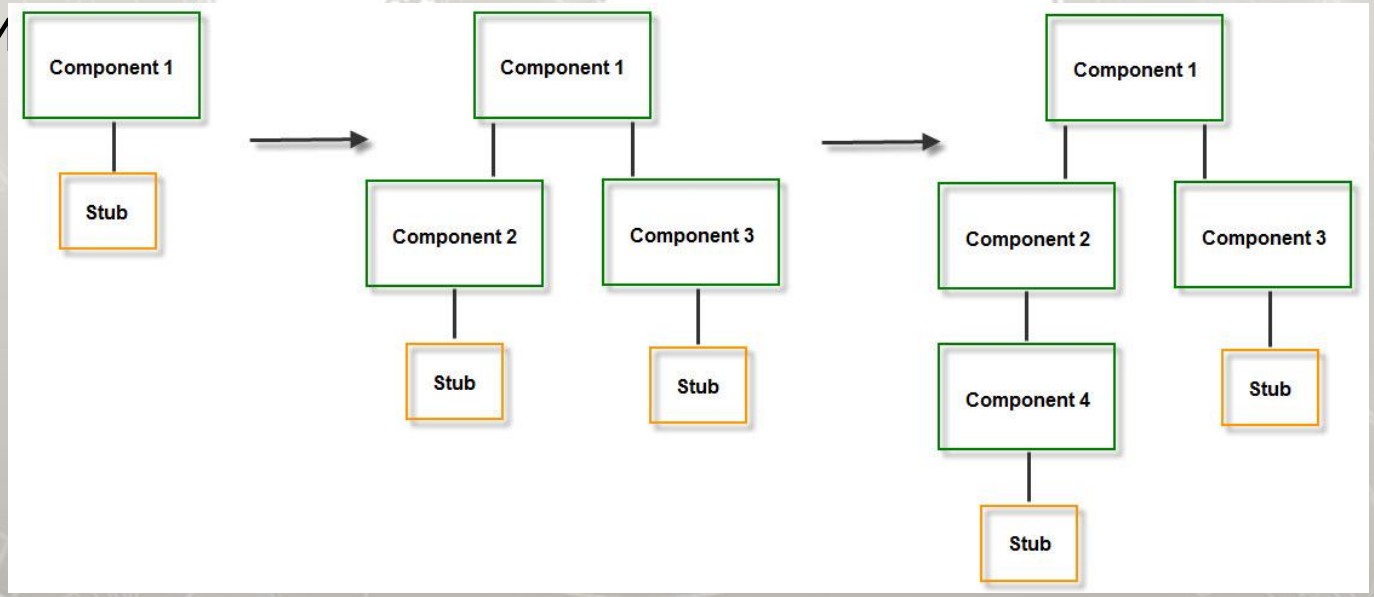
Test Levels

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. Этот подход упрощает локализацию багов, но необходимо создавать драйв



Test Levels

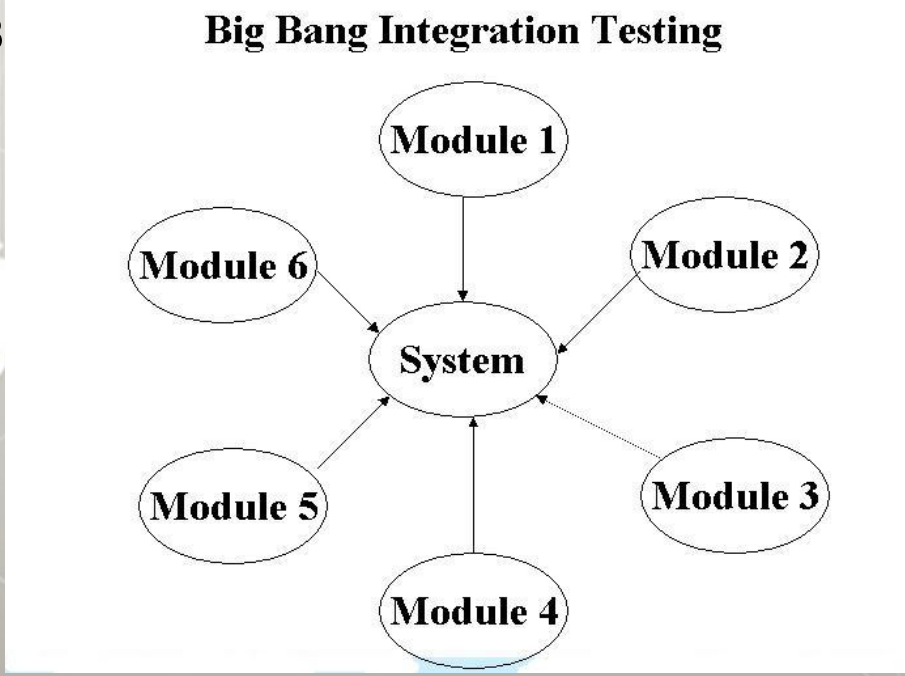
Сверху вниз (***Top Down Integration***) - Вначале тестируются все высокоуровневые модули, и постепенно один за другим добавляются низкоуровневые. Такой подход предполагает следование за разработкой. Все модули более низкого уровня симулируются заглушками с аналогичной функци



Test Levels

Большой взрыв (*"Big Bang" Integration*)

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. Основным преимуществом является отсутствие з
локализация багов
усложняется.



Test Levels

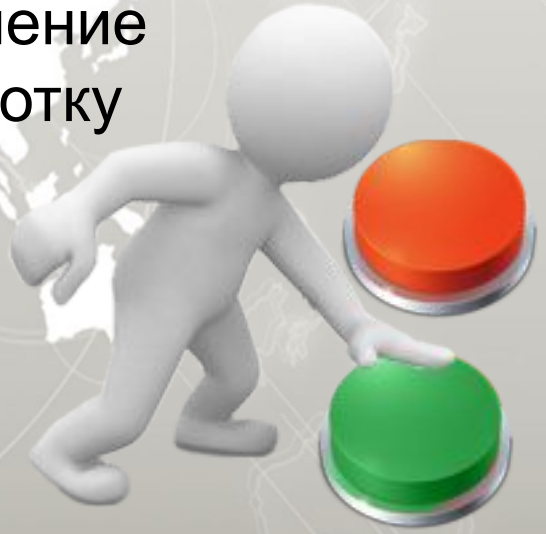
Системное или «энд-ту-энд» тестирование (*system or end-to-end testing*) направлено на проверку всего приложения как единого целого. Здесь появляется возможность полноценно взаимодействовать с приложением с точки зрения конечного пользователя.

Подходы к системному тестированию:

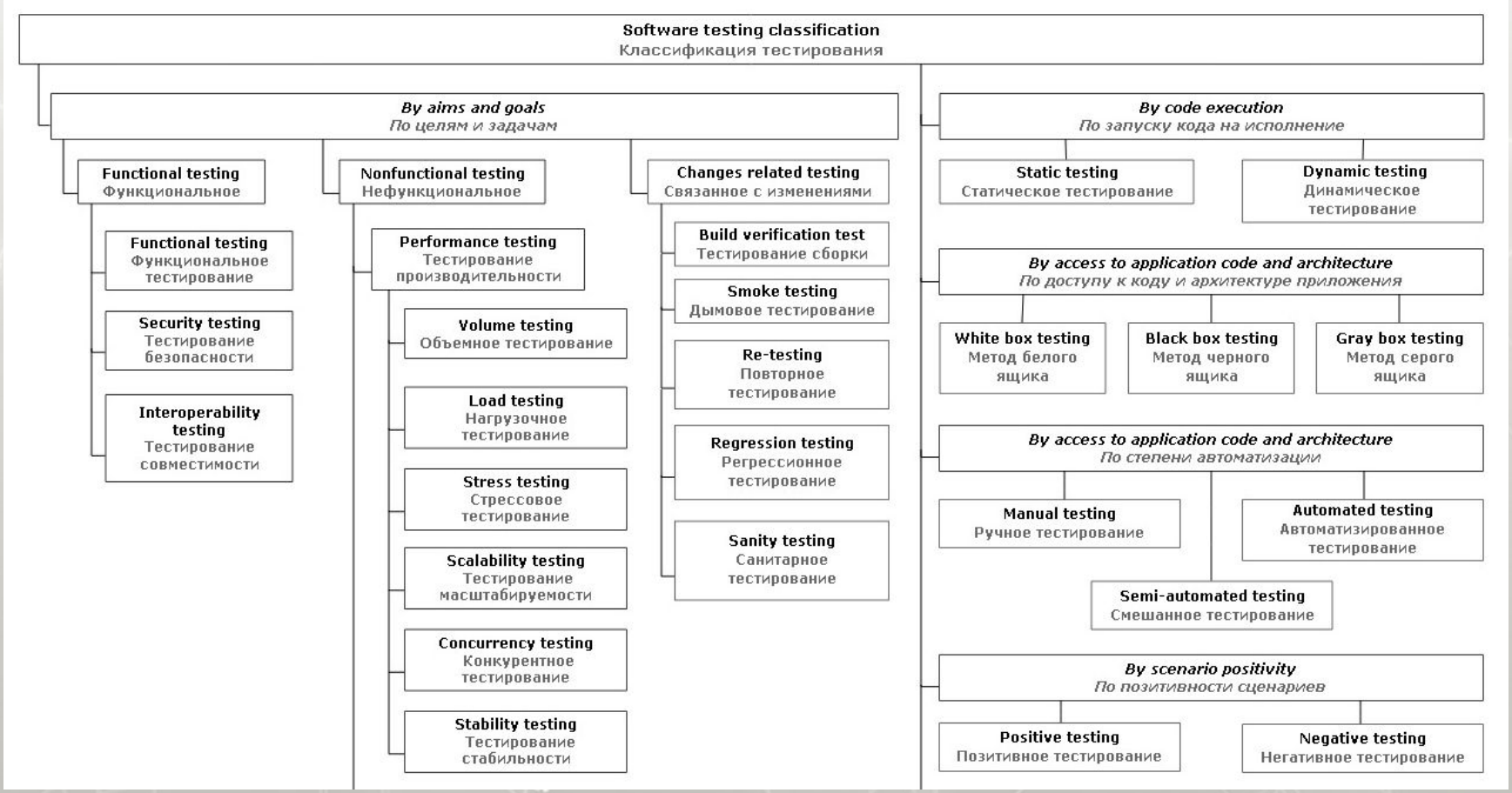
- **на базе требований** (***requirements based***) - для каждого требования пишутся тестовые случаи (test
данного требования.
- на **базе случаев использования** (*use case based*) на основе представления о способах использования продукта создаются случаи использования системы (UseCases).

Test Levels

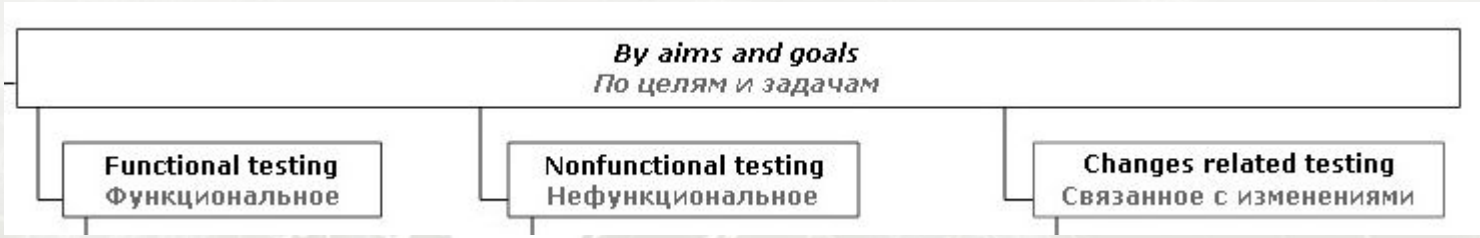
Приёмочное тестирование (*acceptance testing*) — формализованное тестирование, направленное на проверку приложения с точки зрения конечного пользователя/заказчика и вынесения решения о том, принимает ли заказчик работу у исполнителя (проектной команды). Фаза приемочного тестирования длится до тех пор, пока заказчик не выносит решение об отправлении приложения на доработку или выдаче приложения.



Software testing classification



Classification by aims and goals



Функциональные виды тестирования (***functional testing***) базируются на функциях и особенностях, а также взаимодействии с другими системами. *Рассматривают внешнее поведение системы*.

Нефункциональные виды тестирования (***non-functional testing***) - описывает тесты, необходимые для определения нефункциональных особенностей приложения, которые могут быть измерены различными величинами. В целом, это тестирование того, *"Как" система работает*.(насколько удобна, совместима, какова производительность, как учтена защита безопасности и т.д.).

Связанные с изменениями виды тестирования.

Functional tests

Функциональные виды тестирования (***functional testing***) - направлены на обеспечение характеристик качества, включенных в показатель *функциональность*.

- Функциональное тестирование (functional testing)
- Тестирование безопасности (security testing)
- Тестирование совместимости (interoperability testing)

Functional testing

Функциональное тестирование (***functional testing***) – направлено на проверку корректности работы функциональности приложения. Тестирование функциональности может проводиться в двух аспектах:

- Тестирование в перспективе «требования» использует спецификацию функциональных требований к системе как основу для дизайна тестовых случаев (***test cases***).
- Тестирование в перспективе «бизнес-процессы» использует знание этих самых бизнес-процессов, которые описывают сценарии ежедневного использования системы. В этой перспективе тестовые сценарии (***test scripts***), как правило, основываются на случаях использования системы (***use cases***).

Преимущества: имитирует фактическое использование системы;
Недостатки: возможность упущения логических ошибок в программном обеспечении и вероятность избыточного тестирования.

Security and Access Control Testing

Тестирование безопасности (***Security and Access Control Testing***) - тестирование, направленное на проверку способности приложения противостоять злонамеренным попыткам получения доступа к данным или функциям, права на доступ к которым у злоумышленника нет.

Общая стратегия безопасности основывается на трех основных принципах:

- конфиденциальность
- целостность
- доступность

Types of vulnerabilities

Наиболее распространенными видами уязвимости в безопасности программного обеспечения являются:

- XSS (***Cross-Site Scripting***) - на генерированной сервером странице, выполняются вредоносные скрипты, с целью атаки клиента
- XSRF / CSRF (***Request Forgery***) - позволяющий использовать недостатки HTTP протокола
- ***Code injections*** (SQL, PHP, ASP и т.д.) - запуск исполняемого кода с целью получения доступа к системным ресурсам
- ***Server-Side Includes (SSI) Injection*** - вставку серверных команд в HTML код или запуск их напрямую с сервера
- ***Authorization Bypass*** - получить несанкционированный доступ к учетной записи или документам другого пользователя

Interoperability Testing

Тестирование совместимости (***Interoperability Testing***) — тестирование, направленное на проверку способности приложения работать в указанном окружении.

Здесь может проверяться:

- совместимость с аппаратной платформой, операционной системой и сетевой инфраструктурой (***конфигурационное тестирование, configuration testing***)
- совместимость с браузерами и их версиями (***кросс-браузерное тестирование, cross-browser testing***)
- совместимость с мобильными устройствами (***mobile testing***).

Non-functional tests

Нефункциональные виды тестирования (non-functional testing) - определения нефункциональных особенностей приложения, которые могут быть измерены различными величинами ("Как" система работает).

- Тестирование **производительности** (***performance testing***)
- Тестирование **установки** (***installation testing***)
- Тестирование **интерфейса пользователя** (***GUI testing***)
- Тестирование **удобства использования** (***usability testing***)
- Тестирование **восстанавливаемости** (***recovery testing***)
- Тестирование **отказоустойчивости** (***failover testing***)
- Тестирование **интернационализации** (***internationalization testing***)

Performance testing

Тестирование производительности (***performance testing***) — исследование показателей скорости реакции приложения на внешние воздействия при различной по характеру и интенсивности нагрузке (высокая, предельная, стрессовая).

- **Объёмное** тестирование (***volume testing***)
- **Нагрузочное** тестирование (***load testing***)
- **Стрессовое** тестирование (***stress testing***)
- Тестирование **масштабируемости** (***scalability testing***)
- **Конкурентное** тестирование (***concurrency testing***)
- Тестирование **стабильности** (***stability testing***)

Installation testing

Тестирование установки (***installation testing***) — тестирование, направленное на выявление дефектов, влияющих на протекание стадии инсталляции (установки) приложения. В общем случае такое тестирование проверяет множество сценариев и аспектов работы инсталлятора в таких ситуациях, как:

- новая среда исполнения, в которой приложение ранее не было инсталлировано;
- обновление существующей версии («апгрейд»);
- изменение текущей версии на более старую («даунгрейд»);
- повторная установка приложения с целью устранения возникших проблем («переинсталляция»);
- повторный запуск инсталляции после ошибки, приведшей к невозможности продолжения инсталляции;
- удаление приложения;
- установка нового приложения из семейства приложений;
- автоматическая инсталляция без участия пользователя.

User interface (UI) testing

Тестирование интерфейса пользователя (***GUI testing***) – обнаружение ошибок в функциональности посредством интерфейса, необработанные исключения при взаимодействии с интерфейсом, потеря или искажение данных, передаваемых через элементы интерфейса, ошибки в интерфейсе (несоответствие проектной документации, отсутствие элементов интерфейса).

Тестирование удобства использования (***usability testing***) — исследование того, насколько конечному пользователю понятно, как работать с продуктом, а также на то, насколько ему нравится использовать продукт.

!Usabilitytesting и GUItesting не одно и тоже: корректно работающий интерфейс может быть неудобным, а удобный может работать некорректно.

Recovery and Failover testing

Тестирование восстанавливаемости (***recovery testing***) — тестирование способности приложения восстанавливать свои функции и заданный уровень производительности, а также восстанавливать данные в случае возникновения критической ситуации, приводящей к временной (частичной) утрате работоспособности приложения.

Тестирование отказоустойчивости (***failover testing***) — тестирование, заключающееся в эмуляции или реальном создании критических ситуаций с целью проверки способности приложения задействовать соответствующие механизмы, предотвращающие нарушение работоспособности, производительности и повреждения данных.

Program interfaces

Тестирование интернационализации (***internationalization testing***) — тестирование, направленное на проверку готовности продукта к работе с использованием различных языков и с учётом различных национальных и культурных особенностей. Например: что будет, если открыть файл с иероглифом в имени; как будет работать интерфейс, если всё перевести на японский и т.д.

Тестирование локализации (***localization testing***) — тестирование, направленное на проверку корректности и качества адаптации продукта к использованию на том или ином языке с учётом национальных и культурных особенностей.

Changes-based testing

После проведения необходимых изменений, таких как исправление бага/дефекта, программное обеспечение должно быть перетестировано для подтверждения того факта, что проблема была действительно решена.

- **Тестирование сборки** (***build verification test***)
- **Дымовое тестирование** (***smoke testing***)
- **Повторное тестирование** (***re-testing***)
- **Регрессионное тестирование** (***regression testing***)
- **Санитарное тестирование** (***sanity testing***)

Smoke and Sanity testing

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

Санитарное тестирование (***sanity testing***) –узконаправленное тестирование достаточное для доказательства того, что конкретная функция работает согласно заявленным в спецификации требованиям.

Эти виды тестирования имеют "вектора движения", направления в разные стороны. Санитарное тестирование (**Sanity testing**) направлено **вглубь** проверяемой функции, в то время как **дымовое** (**Smoke testing**) направлено **вширь**, для покрытия тестами как можно большего функционала в кратчайшие сроки.

Re-testing and Regression

Повторное тестирование (***re-testing***) — выполнение тест-кейсов, которые ранее обнаружили дефекты, с целью подтверждения устранения дефектов.

Регрессионное тестирование (***regression testing***) — тестирование, направленное на проверку того факта, что в ранее работоспособной функциональности не появились ошибки, вызванные изменениями в приложении или среде его функционирования. Регрессионными могут быть как функциональные, так и нефункциональные тесты.

By code execution

Статическое тестирование (***static testing***) — тестирование без запуска кода на исполнение.
документы (требования, тест-кейсы, описания архитектуры приложения, схемы баз данных и т.д.)
графические прототипы (например, эскизы пользовательского интерфейса)
код приложения (*code review*)
параметры (настройки) среды исполнения приложения.
подготовленные тестовые данные.

Динамическое тестирование (***dynamic testing***) — тестирование с запуском кода на исполнение (проверяется реальное поведение приложения или его части).

By automation level

Ручное тестирование (***manual testing***) — тестирование, в котором тест-кейсы выполняются человеком вручную без использования средств автоматизации

Автоматизированное тестирование (*automated testing*, *test automation*) — набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования.

Смешанное тестирование ***(semi-automated testing)*** - здесь ручной подход сочетается с автоматизированным. Например, с помощью тула я создаю новый эккаунт и потом вручную генерирую транзакцию покупки.

By scenario positivity

Позитивное тестирование (***positive testing***) направлено на исследование приложения в ситуации, когда все действия выполняются строго по инструкции без каких бы то ни было ошибок, отклонений, ввода неверных данных и т.д.

Негативное тестирование (***negative testing***) — направлено на исследование работы приложения в ситуациях, когда с ним выполняются (некорректные) операции и/или используются данные, потенциально приводящие к ошибкам (классика жанра — деление на ноль).

By application nature

Тестирование веб-приложений (***web-applications testing***) - тестирование совместимости (в особенности — кросс-браузерного тестирования), производительности, автоматизации тестирования с использованием широкого спектра инструментальных средств.

Тестирование мобильных приложений (***mobile applications testing***) - повышенное внимание к тестированию совместимости, оптимизации производительности (в том числе клиентской части с точки зрения снижения энергопотребления), автоматизации тестирования с применением эмуляторов мобильных устройств.

*T***естирование настольных приложений** (***desktop applications testing***) является самым классическим среди всех перечисленных в данной классификации, и его особенности зависят от предметной области приложения, нюансов архитектуры, ключевых показателей качества и т.д.

By end users participation

Операционное тестирование (***operational testing***) - проводится в реальной или приближенной к реальной операционной среде, включающей операционную систему, системы управления базами данных, серверы приложений, веб-серверы, аппаратное обеспечение и т.д.

Альфа-тестирование (***alpha testing***) выполняется внутри организации разработчика с возможным частичным привлечением конечных пользователей.

Бета-тестирование (***beta testing***) выполняется вне организации разработчика с активным привлечением конечных пользователей/заказчиков.

Гамма-тестирование (***gamma testing***) — финальная стадия тестирования перед выпуском продукта, направленная на исправление незначительных дефектов, обнаруженных в бета-тестировании.

By formalization level

Тестирование **на основе тест-кейсов** (***test case based testing***) — формализованный подход, в котором тестирование производится на основе заранее подготовленных тест-кейсов, наборов тест-кейсов и иной документации.

Исследовательское тестирование (***exploratory testing***) — частично формализованный подход, в рамках которого тестировщик выполняет работу с приложением по выбранному сценарию (***use case***), который, в свою очередь, дорабатывается в процессе выполнения с целью более полного исследования приложения.

- **Сессионное тестирование** (***session-based testing***) основанный на сценарном подходе
- Тестированием **на основе чек-листов** (***checklist-based testing***) основанный на чек листах

Свободное (интуитивное) тестирование (***ad hoc testing***) — полностью неформализованный подход, тестировщик полностью опирается на свой профессионализм и интуицию (***experience-based testing***) для спонтанного выполнения с приложением действий, которые могут обнаружить ошибку.