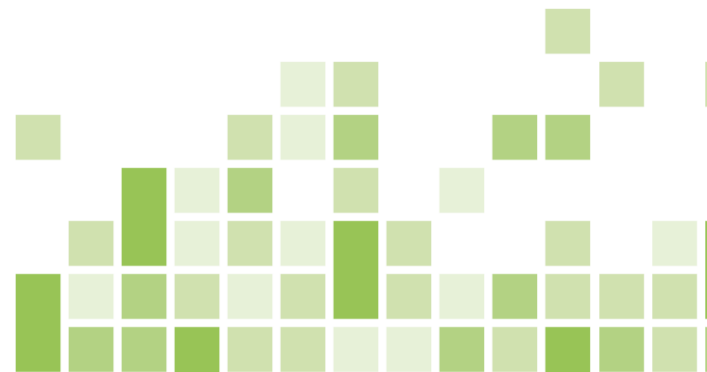


**ТОМСКИЙ
ПОЛИТЕХ**



Тестирование программного обеспечения

Лекция №2.

Классификация тестирования

Копнов Максим Валериевич

**Томск
2021**

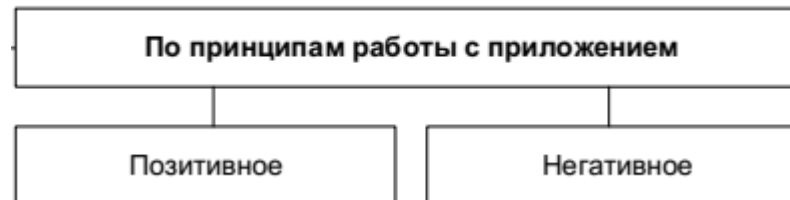
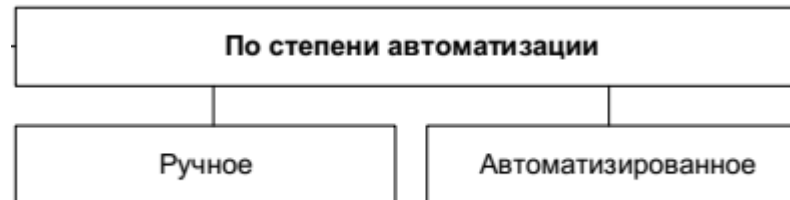
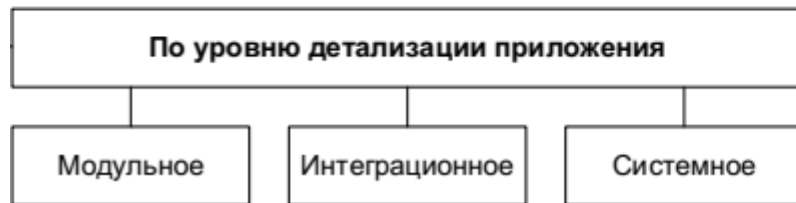
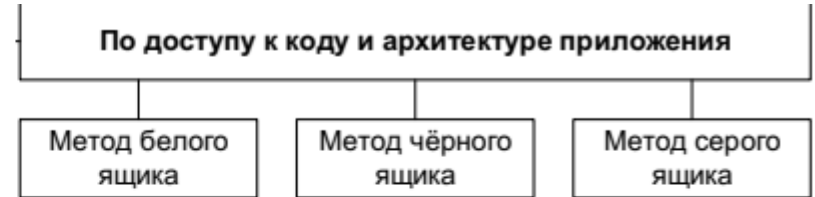
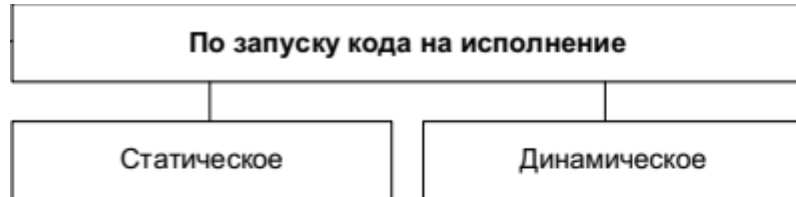
Вспоминаем прошлое занятие

- **Определение тестирования**
- **Определение качества**
- **Определение обеспечения качества**

О чем сегодня будем говорить

- **Классификация тестирования**

Классификация тестирования



Классификация тестирования

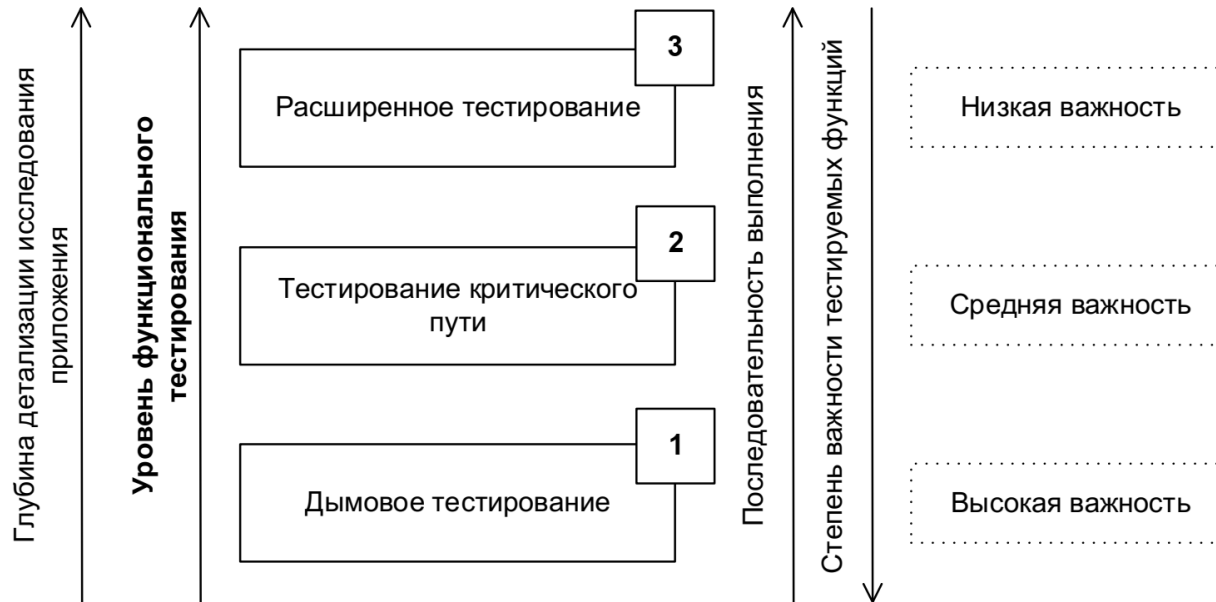
- По запуску кода на исполнение:
 - Статическое тестирование — без запуска.
 - Динамическое тестирование — с запуском.
- По доступу к коду и архитектуре приложения:
 - Метод белого ящика — доступ к коду есть.
 - Метод чёрного ящика — доступа к коду нет.
 - Метод серого ящика — к части кода доступ есть, к части — нет.
- По степени автоматизации:
 - Ручное тестирование — тест-кейсы выполняет человек.
 - Автоматизированное тестирование — тест-кейсы частично или полностью выполняет специальное инструментальное средство.

Классификация тестирования

- По уровню детализации приложения (по уровню тестирования):
 - Модульное (компонентное) тестирование — проверяются отдельные небольшие части приложения.
 - Интеграционное тестирование — проверяется взаимодействие между несколькими частями приложения.
 - Системное тестирование — приложение проверяется как единое целое.
- По уровню функционального тестирования:
 - Дымовое тестирование — проверка самой важной, самой ключевой функциональности, неработоспособность которой делает бессмысленной саму идею использования приложения.
 - Тестирование критического пути — проверка функциональности, используемой типичными пользователями в типичной повседневной деятельности.

Классификация тестирования

- По уровню функционального тестирования:
 - Расширенное тестирование — проверка всей функциональности, заявленной в требованиях.



Классификация тестирования

- По принципам работы с приложением:
 - Позитивное тестирование — все действия с приложением выполняются строго по инструкции без никаких недопустимых действий, некорректных данных и т.д. Можно образно сказать, что приложение исследуется в «тепличных условиях».
 - Негативное тестирование — в работе с приложением выполняются (некорректные) операции и используются данные, потенциально приводящие к ошибкам (деление на ноль)

Классификация тестирования

- По природе приложения:
 - Тестирование веб-приложений — тестирование совместимости (кросс-браузерное тестирование), производительности.
 - Тестирование мобильных приложений — тестирование совместимости (кроссплатформенное, кросс форматное тестирование), производительности.
 - Тестирование настольных приложений — особенности этого тестирования зависят от предметной области приложения, нюансов архитектуры, ключевых показателей качества и т.д.

Классификация тестирования

- По уровню архитектуры приложения:
 - Тестирование уровня представления — исследуются вопросы удобства использования, скорости отклика интерфейса, совместимости с браузерами, корректности работы интерфейсов.
 - Тестирование уровня бизнес-логики — отвечает за проверку основного набора функций приложения и строится на базе ключевых требований к приложению, бизнес-правил и общей проверки функциональности.
 - Тестирование уровня данных — тестирование данных, проверка соблюдения бизнес-правил, тестирование производительности.

Классификация тестирования

- По уровню архитектуры приложения:
 - Тестирование уровня представления — исследуются вопросы удобства использования, скорости отклика интерфейса, совместимости с браузерами, корректности работы интерфейсов.
 - Тестирование уровня бизнес-логики — отвечает за проверку основного набора функций приложения и строится на базе ключевых требований к приложению, бизнес-правил и общей проверки функциональности.
 - Тестирование уровня данных — тестирование данных, проверка соблюдения бизнес-правил, тестирование производительности.

Классификация тестирования

- По привлечению конечных пользователей:
 - Альфа-тестирование — выполняется внутри организации-разработчика с возможным частичным привлечением конечных пользователей.
 - Бета-тестирование — выполняется вне организации-разработчика с активным привлечением конечных пользователей/заказчиков.
 - Гамма-тестирование — финальная стадия тестирования перед выпуском продукта, направленная на исправление незначительных дефектов, обнаруженных в бета-тестировании. Как правило, также выполняется с максимальным привлечением конечных пользователей/заказчиков.

Классификация тестирования

- По степени формализации:
 - Тестирование на основе тест-кейсов — формализованный подход, в котором тестирование производится на основе заранее подготовленных тест-кейсов, наборов тест-кейсов и иной документации.
 - Исследовательское тестирование — частично формализованный подход, в рамках которого тестировщик выполняет работу с приложением по выбранному сценарию, который, в свою очередь, дорабатывается в процессе выполнения с целью более полного исследования приложения.
 - Свободное тестирование — полностью неформализованный подход, в котором не предполагается использования ни тест-кейсов, ни чек-листов, ни сценариев — тестировщик полностью опирается на свой профессионализм и интуицию для спонтанного выполнения с приложением действий, которые, как он считает, могут обнаружить ошибку.

Классификация тестирования по целям и задачам

- По целям и задачам:
 - Позитивное тестирование — все действия с приложением выполняются строго по инструкции без никаких недопустимых действий, некорректных данных и т.д. Можно образно сказать, что приложение исследуется в «тепличных условиях».
 - Негативное тестирование — в работе с приложением выполняются (некорректные) операции и используются данные, потенциально приводящие к ошибкам (деление на ноль).
 - Функциональное тестирование — направлено на проверку корректности работы функциональности приложения корректность реализации функциональных требований.
 - Нефункциональное тестирование — выполняет проверку нефункциональных особенностей приложения (корректность реализации нефункциональных требований, таких как удобство использования, совместимость, производительность, безопасность и т.д.

Классификация тестирования по целям и задачам

- Инсталляционное тестирование — направленно на выявление дефектов при установке приложения. Проверяются различные сценарии установки. Обновления и удаления приложения:
 - 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.

Классификация тестирования по целям и задачам

- Регрессионное тестирование — тестирование, направленное на проверку того факта, что в ранее работоспособной функциональности не появились ошибки, вызванные изменениями в приложении или среде его функционирования.
- Повторное тестирование — выполнение тест-кейсов, которые ранее обнаружили дефекты, с целью подтверждения устранения дефектов.
- Операционное тестирование — тестирование, проводимое в реальной или приближенной к реальной операционной среде, включающей операционную систему, системы управления базами данных, серверы приложений, веб-серверы, аппаратное обеспечение и т.д.
- Тестирование доступности — тестирование, направленное на исследование пригодности продукта к использованию людьми с ограниченными возможностями (слабым зрением и т.д.).

Классификация тестирования по целям и задачам

- Приёмочное тестирование — формализованное тестирование, направленное на проверку приложения с точки зрения конечного пользователя/заказчика и вынесения решения о том, принимает ли заказчик работу у исполнителя. Подвиды приёмочного тестирования:
 - Производственное приёмочное тестирование — выполняемое проектной командой исследование полноты и качества реализации приложения с точки зрения его готовности к передаче заказчику.
 - Операционное приёмочное тестирование — операционное тестирование, выполняемое с точки зрения выполнения инсталляции, потребления приложением ресурсов, совместимости с программной и аппаратной платформой и т.д.
 - Итоговое приёмочное тестирование — тестирование конечными пользователями приложения в реальных условиях эксплуатации с целью вынесения решения о том, требует ли приложение доработок или может быть принято в эксплуатацию в текущем виде.

Классификация тестирования по целям и задачам

- Демонстрационное тестирование— формальный процесс демонстрации заказчику продукта с целью подтверждения, что продукт соответствует всем заявленным требованиям.
- Сравнительное тестирование — сравнительный анализ преимуществ и недостатков разрабатываемого продукта по отношению к его основным конкурентам.
- Исчерпывающее тестирование — тестирование приложения со всеми возможными комбинациями всех возможных входных данных во всех возможных условиях выполнения.
- Тестирование надёжности — тестирование способности приложения выполнять свои функции в заданных условиях на протяжении заданного времени или заданного количества операций.

Классификация тестирования по целям и задачам

- Тестирование восстанавливаемости — 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 — тестирование, заключающееся в эмуляции или реальном создании критических ситуаций с целью проверки способности приложения задействовать соответствующие механизмы, предотвращающие нарушение работоспособности, производительности и повреждения данных.

Классификация тестирования по целям и задачам

- Тестирование удобства использования — исследование того, насколько конечному пользователю понятно, как работать с продуктом, а также, насколько ему нравится использовать продукт.
- Тестирование интерфейса — проверка интерфейсов приложения или его компонентов.
- Тестирование безопасности — проверка способности приложения противостоять злонамеренным попыткам получения доступа к данным или функциям, права на доступ к которым у злоумышленника нет.
- Тестирование интернационализации — проверка готовности продукта к работе с использованием различных языков и с учётом различных национальных и культурных особенностей.
- Тестирование локализации — проверку корректности и качества адаптации продукта к использованию на том или ином языке с учётом национальных и культурных особенностей.

Классификация тестирования по целям и задачам

- Тестирование совместимости — тестирование, направленное на проверку способности приложения работать в указанном окружении. Проверяется:
 - Совместимость с аппаратной платформой, операционной системой и сетевой инфраструктурой.
 - Совместимость с браузерами и их версиями.
 - Совместимость с мобильными устройствами.
 - ...
- Тестирование данных и баз данных — исследование таких характеристик данных, как полнота, непротиворечивость, целостность, структурированность и т.д.
- Тестирование использования ресурсов — совокупность видов тестирования, проверяющих эффективность использования приложением доступных ему ресурсов и зависимость результатов работы приложения от количества доступных ему ресурсов.

Классификация тестирования по целям и задачам – Тестирование производительности

- Тестирование производительности — исследование показателей скорости реакции приложения на внешние воздействия при различной по характеру и интенсивности нагрузке. Подвиды тестирования производительности:
 - Нагрузочное тестирование — исследование способности приложения сохранять заданные показатели качества при нагрузке в допустимых пределах и некотором превышении этих пределов.
 - Тестирование масштабируемости — исследование способности приложения увеличивать показатели производительности в соответствии с увеличением количества доступных приложению ресурсов.
 - Объёмное тестирование — исследование производительности приложения при обработке различных (как правило, больших) объёмов данных.
 - Стрессовое тестирование — исследование поведения приложения при нештатных изменениях нагрузки, значительно превышающих расчётный уровень, или в ситуациях недоступности значительной части необходимых приложению ресурсов.

Классификация тестирования по целям и задачам – Тестирование производительности

- Конкурентное тестирование — исследование поведения приложения в ситуации, когда ему приходится обрабатывать большое количество одновременно поступающих запросов, что вызывает конкуренцию между запросами за ресурсы (базу данных, память, канал передачи данных, дисковую подсистему и т.д.).

Классификация по степени вмешательства в работу приложения

- Инвазивное тестирование — тестирование, выполнение которого может повлиять на функционирование приложения в силу работы инструментов тестирования (например, будут искажены показатели производительности) или в силу вмешательства в сам код приложения
- Неинвазивное тестирование — тестирование, выполнение которого незаметно для приложения и не влияет на процесс его обычной работы.

Классификация по техникам автоматизации

- Тестирование под управлением данными — способ разработки автоматизированных тест-кейсов, в котором входные данные и ожидаемые результаты выносятся за пределы тест-кейса и хранятся вне его — в файле, базе данных и т.д.
- Тестирование под управлением ключевыми словами — способ разработки автоматизированных тест-кейсов, в котором за пределы тест-кейса выносятся не только набор входных данных и ожидаемых результатов, но и логика поведения тест-кейса, которая описывается ключевыми словами (командами).
- Тестирование под управлением поведением (behavior-driven testing) — способ разработки автоматизированных тест-кейсов, в котором основное внимание уделяется корректности работы бизнес-сценариев, а не отдельным деталям функционирования приложения

Классификация на основе (знания) источников ошибок

- Тестирование предугадыванием ошибок — тесты разрабатываются на основе опыта тестировщика и его знаний о том, какие дефекты типичны для тех или иных компонентов или областей функциональности приложения.
- Эвристическая оценка — техника, направленная на поиск проблем в интерфейсе пользователя, представляющих собой отклонение от общепринятых норм.
- Мутационное тестирование — сравнивается поведение нескольких версий одного и того же компонента, причём часть таких версий может быть специально разработана с добавлением ошибок.
- Тестирование добавлением ошибок — техника тестирования, в которой в приложение специально добавляются заранее известные, специально продуманные ошибки с целью мониторинга их обнаружения и устранения и, таким образом, формирования более точной оценки показателей процесса тестирования.

Классификация на основе выбора входных данных

- Тестирование на основе классов эквивалентности — направленно на сокращение количества разрабатываемых и выполняемых тест-кейсов при сохранении достаточного тестового покрытия.
- Тестирование на основе граничных условий — инструментальная техника тестирования на основе классов эквивалентности, позволяющая выявить специфические значения исследуемых параметров, относящиеся к границам классов эквивалентности.
- Доменное тестирование — техника тестирования на основе классов эквивалентности и граничных условий, позволяющая эффективно создавать тест-кейсы, затрагивающие несколько параметров (переменных) одновременно (в том числе с учётом взаимозависимости этих параметров).
- Парное тестирование — техника тестирования, в которой тест-кейсы строятся по принципу проверки пар значений параметров (переменных) вместо того, чтобы пытаться проверить все возможные комбинации всех значений всех параметров.

Классификация на основе выбора входных данных

- Тестирование на основе ортогональных массивов — инструментальная техника попарного и N-комбинационного тестирования, основанная на использовании т.н. «ортогональных массивов» (двумерных массивов, обладающих следующим свойством: если взять две любые колонки такого массива, то получившийся «подмассив» будет содержать все возможные попарные комбинации значений, представленных в исходном массиве).



ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

Тестирование программного обеспечения

Копнов Максим Валериевич
kopnovmv@tpu.ru

