**Список основных определений**

|  |  |  |
| --- | --- | --- |
| **Термин (по- русски)** | **Термин (по- английски)** | **Определение** |
| Автоматизиро- ванное тести- рование{73} | Automated test- ing | Набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе  тестирования. |
| Артефакты Тестирования | Testing Artifacts | Тест план;  Тест кейс,Тестовый сценарий,Чек-лист;  Баги;  Отчёты по тестированию.**Тест сьют (Test Script or Test Suite)** |
| Альфа-тести- рование{81} | Alpha testing | Тестирование, которое выполняется внутри организации-разработчика с возможным ча- стичным привлечением конечных пользова- телей. Может являться формой внутреннего  приёмочного тестирования. |
| Анализ перво- причин{250} | Root cause analysis | Процесс исследования и классификации пер- вопричин возникновения событий, негативно влияющих на безопасность, здоровье, окру- жающую среду, качество, надёжность и про- изводственный процесс. |
| Бета-тестиро- вание{81} | Beta testing | Тестирование, которое выполняется вне ор- ганизации-разработчика с активным привле-  чением конечных пользователей/заказчиков. |
| Граничное условие{234} | Border condi- tion, boundary  condition | Значение, находящееся на границе классов эквивалентности. |
| Дефект{166} | Defect, anomaly | Отклонение фактического результата от ожи- даний наблюдателя, сформированных на ос- нове требований, спецификаций, иной доку- ментации или опыта и здравого смысла. |
| Динамическое  тестирова- ние{70} | Dynamic testing | Тестирование с запуском кода на исполне- ние. |
| Дымовое те- стирование{76} | Smoke test | Тестирование, которое направлено на про- верку самой главной, самой важной, самой ключевой функциональности, неработоспо- собность которой делает бессмысленной саму идею использования приложения (или иного объекта, подвергаемого дымовому те-  стированию). |

|  |  |  |
| --- | --- | --- |
| Инспекция (аудит) кода{94} | Code review, code inspection | Семейство техник повышения качества кода за счёт того, что в процессе создания или со- вершенствования кода участвуют несколько человек. В отличие от техник статического анализа кода (по потоку управления и потоку данных), аудит кода также улучшает такие его характеристики как понятность, поддер- живаемость, соответствие соглашениям об оформлении и т.д. Аудит кода выполняется в  основном самими программистами. |
| Интеграцион- ное тестирова- ние | Integration test- ing | Тестирование, которое направлено на про- верку взаимодействия между несколькими частями приложения (каждая из которых, в свою очередь, проверена отдельно на стадии модульного тестирования). |
| Класс эквива- лентности{234} | Equivalence class | Набор данных, обрабатываемых одинаковым образом и приводящих к одинаковому ре- зультату. |
| Метод белого ящика{70} | White box test- ing | Метод тестирования, в рамках которого у те- стировщика есть доступ к внутренней струк- туре и коду приложения, а также есть доста- точно знаний для понимания увиденного. |
| Метод серого ящика{71} | Gray box testing | Комбинация методов белого ящика и чёрного ящика, состоящая в том, что к части кода и архитектуры у тестировщика доступ есть, а к части — нет. (См. пояснения по альтернатив- ному определению здесь: {71}.) |
| Метод чёрного ящика{71} | Black box test- ing | Метод тестирования, в рамках которого у те- стировщика либо нет доступа к внутренней структуре и коду приложения, либо недоста- точно знаний для их понимания, либо он со-  знательно не обращается к этим данным в процессе тестирования. |
| Метрика{210} | Metric | Числовая характеристика показателя каче- ства. Может включать описание способов  оценки и анализа результата. |
| Модель разра- ботки ПО{18} | Software Devel- opment Model | Структура, систематизирующая различные виды проектной деятельности, их взаимодей- ствие и последовательность в процессе раз- работки ПО. Выбор той или иной модели за- висит от масштаба и сложности проекта, предметной области, доступных ресурсов и  множества других факторов. |
| Модульное (компонентное)  тестирова- ние{74} | Unit testing, component test- ing | Тестирование, направленное на проверку от- дельных небольших частей приложения, ко- торые (как правило) можно исследовать изо- лированно от других подобных частей. |
| Набор тест- кейсов{143} | Test case suite, test suite, test set | Совокупность тест-кейсов, выбранных с неко- торой общей целью или по некоторому об- щему признаку. |

|  |  |  |
| --- | --- | --- |
| Негативное те- стирование{79} | Negative testing | Тестирование, направленное на исследова- ние работы приложения в ситуациях, когда с ним выполняются (некорректные) операции и/или используются данные, потенциально  приводящие к ошибкам. |
| Нефункцио- нальное тести- рование{83} | Non-functional testing | Тестирование, направленное на проверку не- функциональных особенностей приложения (корректность реализации нефункциональ- ных требований), таких как удобство исполь- зования, совместимость, производитель- ность, безопасность и т.д. |
| Нефункцио- нальные требо- вания{38} | Non-functional requirements | Требования, описывающие свойства системы (удобство использования, безопасность, надёжность, расширяемость и т.д.), кото- рыми она должна обладать при реализации  своего поведения. |
| Отчёт о де- фекте{167} | Defect report | Документ, описывающий и приоритизирую- щий обнаруженный дефект, а также содей- ствующий его устранению. |
| Отчёт о ре- зультатах те- стирования{217} | Test progress report, test sum- mary report | Документ, обобщающий результаты работ по тестированию и содержащий информацию, достаточную для соотнесения текущей ситуа- ции с тест-планом и принятия необходимых управленческих решений. |
| Отчётность{206} | Reporting | Сбор и распространение информации о ре- зультатах работы (включая текущий статус, оценку прогресса и прогноз развития ситуа-  ции). |
| Планирова- ние{206} | Planning | Непрерывный процесс принятия управленче- ских решений и методической организации усилий по их реализации с целью обеспече- ния качества некоторого процесса на протя- жении длительного периода времени. |
| Позитивное те- стирование{79} | Positive testing | Тестирование, направленное на исследова- ние приложения в ситуации, когда все дей- ствия выполняются строго по инструкции без каких бы то ни было ошибок, отклонений, ввода неверных данных и т.д. |
| Покрытие{212} | Coverage | Процентное выражение степени, в которой исследуемый элемент затронут соответству-  ющим набором тест-кейсов. |
| Приёмочное тестирова- ние{84} | Acceptance testing | Формализованное тестирование, направлен- ное на проверку приложения с точки зрения конечного пользователя/заказчика и вынесе- ния решения о том, принимает ли заказчик работу у исполнителя (проектной команды). |
| Расширенное тестирова- ние{78} | Extended test | Тестирование, направленное на исследова- ние всей заявленной в требованиях функцио-  нальности — даже той, которая низко про- ранжирована по степени важности. |

|  |  |  |
| --- | --- | --- |
| Регрессионное тестирова- ние{84} | Regression test- ing | Тестирование, направленное на проверку того факта, что в ранее работоспособной функциональности не появились ошибки, вы- званные изменениями в приложении или  среде его функционирования. |
| Ручное тести- рование{72} | Manual testing | Тестирование, в котором тест-кейсы выпол- няются человеком вручную без использова- ния средств автоматизации. |
| Системное те- стирование{75} | System testing | Тестирование, направленное на проверку всего приложения как единого целого, со- бранного из частей, проверенных на стадиях  модульного и интеграционного тестирования. |
| Статическое тестирова- ние{70} | Static testing | Тестирование без запуска кода на исполне- ние. |
| Структурная декомпози- ция{227} | Work break- down structure, WBS | Иерархическая декомпозиция объёмных за- дач на всё более и более малые подзадачи с целью упрощения оценки, планирования и мониторинга выполнения работы. |
| Тест{117} | Test | Набор из одного или нескольких тест-кейсов. |
| Тестирование критического пути{77} | Critical path test | Тестирование, направленное на исследова- ние функциональности, используемой типич- ными пользователями в типичной повседнев- ной деятельности. |
| Тестирование под управле- нием дан- ными{90} | Data-driven testing | Способ разработки автоматизированных тест-кейсов, в котором входные данные и ожидаемые результаты выносятся за пре- делы тест-кейса и хранятся вне его — в  файле, базе данных и т.д. |
| Тестирование под управле- нием ключе- выми сло- вами{90} | Keyword-driven testing | Способ разработки автоматизированных  тест-кейсов, в котором за пределы тест-кейса выносится не только набор входных данных и ожидаемых результатов, но и логика пове- дения тест-кейса, которая описывается клю- чевыми словами (командами). |
| Тестирование под управле- нием поведе- нием{90} | Behavior-driven testing | Способ разработки автоматизированных тест-кейсов, в котором основное внимание уделяется корректности работы бизнес-сце- нариев, а не отдельным деталям функциони- рования приложения. |
| Тестирование программного обеспечения{6} | Software testing | Процесс анализа программного средства и сопутствующей документации с целью выяв-  ления дефектов и повышения качества про- дукта. |
| Тестирование производитель- ности{88} | Performance testing | Исследование показателей скорости реакции приложения на внешние воздействия при различной по характеру и интенсивности нагрузке. |

|  |  |  |
| --- | --- | --- |
| Тест-кейс{117} | Test case | Набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства. Под тест- кейсом также может пониматься соответству- ющий документ, представляющий формаль-  ную запись тест-кейса. |
| Тест-план{208} | Test plan | Документ, описывающий и регламентирую- щий перечень работ по тестированию, а также соответствующие техники и подходы, стратегию, области ответственности, ре- сурсы,риски, расписание и ключевые даты. |
| Требование{29} | Requirement | Описание того, какие функции и с соблюде- нием каких условий должно выполнять при- ложение в процессе решения полезной для  пользователя задачи. |
| Трудоза- траты{225} | Man-hours | Количество рабочего времени, необходимого для выполнения работы (выражается в чело- веко-часах). |
| Функциональ-  ная декомпози- ция{267} | Functional de- composition | Процесс определения функции через её раз- деление на несколько низкоуровневых под- функций. |
| Функциональ- ное тестирова- ние{82} | Functional test- ing | Тестирование, направленное на проверку корректности работы функциональности при- ложения (корректность реализации функцио- нальных требований). |
| Функциональ- ные требова-  ния{38} | Functional re- quirements | Требования, описывающие поведение си- стемы, т.е. её действия (вычисления, преоб-  разования, проверки, обработку и т.д.). |
| Чек-лист{112} | Checklist | Набор идей [тест-кейсов]. Последнее слово не зря взято в скобки, т.к. в общем случае чек-лист — это просто набор идей: идей по тестированию, идей по разработке, идей по планированию и управлению — **любых** идей. |

***CONTINUOUS INTEGRATION***

***Оригинал****:* Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily — leading to multiple integrations per day. Each integration is veriﬁed by an automated build (including test) to detect integration errors as quickly as possible.

*Martin Fowler*

***Вольный перевод****:* Continuous Integration (далее — CI) — это практика разработки ПО, при которой участники команды интегрируют изменения настолько часто, насколько это возможно (как минимум, ежедневно или несколько раз в день). Каждая интеграция верифицируется автоматической сборкой (включая тесты) для определения ошибок интеграции настолько быстро, насколько это возможно. *Мартин Фаулер*

**CONTINUOUS DELIVERY**

Continuous Delivery (непрерывная поставка) подразумевает ещё один шаг вперёд в вопросе автоматизации.

Мы **доверяем автотестам настолько**, что на базе их результатов автоматически **формируем и выкладываем релиз** (например, публикуем библиотеку в Maven Central) или внутреннем репозитории.

Затем его (релиз) можно подвергать исследовательскому тестированию, тестированию безопасности и т.д., но ключевое — мы **формируем релиз на базе решения автотестов**.

**CONTINUOUS DEPLOYMENT**

Continuous Deployment (непрерывное развёртывание) подразумевает ещё один шаг вперёд в вопросе автоматизации (ещё дальше).

Мы уже **доверяем автотестам настолько**, что на базе их результатов автоматически **выкатываем релиз на Production**!

Т.е. мы, фактически, исключаем ручное тестирование (его можно проводить уже на бою).

Благодаря этому, мы можем внедрять хоть по 10-20 новых функций (feature — фич) в день.

**А**

Автоматизация тестирования (англ. Test Automation) — Использование программного обеспечения для осуществления или помощи в проведении определенных тестовых процессов, например, управление тестированием, проектирование тестов, выполнение тестов и проверка результатов.

Авторизация является функцией определения прав доступа к ресурсам и управления этим доступом. Авторизация — это не то же самое что идентификация и аутентификация: идентификация — это называние лицом себя системе; аутентификация — это установление соответствия лица названному им идентификатору; а авторизация — предоставление этому лицу возможностей в соответствие с положенными ему правами или проверка наличия прав при попытке выполнить какое-либо действие. Например, авторизацией являются лицензии на осуществление определённой деятельности.

Адаптивная верстка (Adaptive layout) – верстка, при которой шаблоны верстки и функционал для различных устройств с разной шириной экрана создаются и определяются заранее. Сервер будет определять устройство пользователя и выводить нужный шаблон и контент.

Адрес веб-страницы (URL – Uniform Resource Locator) – уникальный адрес веб-страницы или какого-то другого ресурса в Интернете.

Альфа-тестирование — (alpha testing): Моделируемое или действительное эксплуатационное тестирование потенциальными пользователями/заказчиками или независимой командой тестирования на стороне разработчиков, но вне разрабатывающей организации. Альфатестирование часто применяется к коробочному программному обеспечению в качестве внутреннего приёмочного тестирования.

Анализ граничных значений (англ. Boundary Value Analysis — BVA). Разработка тестов методом черного ящика, при котором тестовые сценарии проектируются на основании граничных значений.

Анализатор трафика, или сниффер (от англ. to sniff — нюхать) — программа или устройство для перехвата и анализа сетевого трафика (своего и/или чужого).

Аутентификация (от греческого: αυθεντικός ; реальный или подлинный): подтверждение подлинности чего-либо или кого либо. Например, предъявление паспорта - это подтверждение подлинности заявленного имени отчества.

**Б**

Баг (дефект) — Изъян в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию, например неверный оператор или определение данных. Дефект, обнаруженный во время выполнения, может привести к отказам компонента или системы. Приоритет багов — важность той или иной ошибки в ПО: Trivial — косметическая малозаметная проблема. Minor — очевидная, незначительная проблема. Major — значительная проблема. Critical — проблема, нарушающая работу c ключевыми функциями ПО. Blocker — проблема, нарушающая функционирование ПО.

Баг-репорт — документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

Баннер — графическое изображение рекламного характера. Баннеры размещают для привлечения клиентов или для формирования позитивного имиджа.

Бекэнд (backend) — область веб-технологий, работающих на сервере. Внутренняя часть серверной системы, занимается обработкой данных.

Белый ящик (англ. White box) — Процедура разработки или выбора тестовых сценариев на основании анализа внутренней структуры компонента или системы.

Бета-тестирование (beta testing) — Эксплутационное тестирование потенциальными и/или существующими клиентами/заказчиками на внешней стороне никак не связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы. Бета-тестирование часто проводится как форма внешнего приёмочного тестирования готового программного обеспечения для того чтобы получить отзывы рынка.

Билд (build, сборка, релиз) — сборка программного обеспечения, имеющая определенную версию. Разработчик выпускает новую версию ПО с новыми функциями или исправлениями. Весь сделанный к этому моменту код помечается (версионируется), архивируется, тестируется и собирается (build) в законченный продукт. Это и есть билд.

Биллинг (billing) - Система выставления счетов клиентам за оказываемые компанией услуги и продаваемые товары.

Браузер (веб-обозреватель) (Browser) — программное обеспечение, предназначенное для просмотра веб-сайтов в интернете. Также может быть использован для просмотра файлов, содержимого веб-документов. К тому же многие современные браузеры поддерживают возможность просмотра документов многих графических форматов, проигрывания аудио и видеофайлов.

Буфер обмена - Место в оперативной памяти компьютера, предназначенное для промежуточного хранения любого вида информации (текст или часть текста, фото и видеофайлы, аудиофайлы и прочие типы файлов).

Бэклог продукта (Резерв продукта, Product backlog)– в проекте гибкой разработки, распределенный по приоритетам список планируемых задач проекта. Резерв может содержать пользовательские истории, бизнес-процессы, запросы на изменение и разработку инфраструктуры. Рабочие элементы из резерва назначаются на будущие итерации на основе их приоритетов.

**В**

Валидация —Доказанное объективными результатами исследования подтверждение того, что требования для ожидаемого конкретного использования приложения были выполнены

Верификация — Доказанное объективными результатами исследования подтверждение того, что определенные требования были выполнены.

Владелец продукта (Product Owner) – представляет интересы конечных пользователей и других заинтересованных в продукте сторон.

Вьюпорт (viewport) — видимая на экране область документа.

**Г**

Гайд (Guide) — у этого слова много синонимов: справочник, руководство, пособие, план, инструкция, путеводитель.

Гибкая методология разработки программного обеспечения (agile software development): Группа методологий разработки программного обеспечения, основанных на итеративной поэтапной разработке, где требования и решения развиваются посредством сотрудничества между самоорганизующимися межфункциональными командами.

Гибкое тестирование (agile testing): Способ тестирования для проектов, использующих гибкие методологии разработки программного обеспечения, включающий такие техники и методы, как экстремальное программирование, и рассматривающий процесс разработки как потребителя процесса тестирования и делающий упор на парадигму раннего тестирования.

Гиперссылка (hyperlink): Указатель на веб-странице, ведущий на другие веб-страницы

Гипертекст — набор (система) текстовых страниц, имеющих перекрестные ссылки. Яркими примерами таких систем являются веб-сайты. В рамках веб-сайтов вы можете переходить по ссылкам с одной страницы на другую. В интернете текст размечается при помощи HTML и изначально предполагает использование гиперссылок.

**Д**

Декомпозиция — разделение одной объёмной сложной задачи на ряд более простых и менее объёмных взаимосвязанных задач.

Динамическое тестирование (англ. Dynamic testing) — тестирование с запуском кода на исполнение.

Драйвер (driver): Компонент программного обеспечения или средство тестирования, которое заменяет компонент, обеспечивающий управление и/или вызов компонента или системы

Дымовое тестирование (англ. Smoke test) — короткий цикл тестов для подтверждения, что после сборки кода (нового или исправленного) приложение стартует и выполняет основные функции.

**З**

Заглушка (placeholder) — текст, отображаемый в поле ввода формы на сайте до того, как пользователь начинает заполнение этого поля. Обычно используется для подсказки о формате ввода данных. Реализуется при помощи атрибута placeholder для тега input.

**И**

Идентификация (от латинского identifico — отождествлять): присвоение субъектам и объектам идентификатора и / или сравнение идентификатора с перечнем присвоенных идентификаторов. Например, представление человека по имени отчеству - это идентификация.

Иконка — миниатюрное изображение, представляющее приложение, документ, окно, компонент ОС, при нажатии на которую открывается соответствующий файл. Необходима для того, чтобы получить быстрый доступ к запрашиваемому файлу.

Инпут (input) – поле для ввода текста или иной элемент формы на веб-странице.

Инсталляционное тестирование (Installation Testing) — проверка успешной инсталляции и настройки, а также обновления или удаления программного обеспечения.

Инструмент захвата/воспроизведения (capture/playback tool): Инструмент выполнения тестов, в котором входная информация записывается во время ручного тестирования с целью создания автоматизированного сценария тестирования, который может быть выполнен позже. Эти средства часто используют для поддержки автоматизированного регрессионного тестирования.

Инструмент нагрузочного тестирования (load testing tool): Инструмент для поддержки нагрузочного тестирования, способный эмулировать увеличивающуюся нагрузку (число одновременных пользователей и/или транзакций во время определенного промежутка времени)

Инструмент отладки (debugging tool): Инструмент, используемый программистами для воспроизведения отказов, исследования состояния программ и поиска соответствующего дефекта. Отладчики позволяют программистам исполнять программу пошагово для останова на любом операторе программы и для установки и проверки программных переменных.

Интеграционное тестирование (integration testing) — Тестирование, выполняемое для обнаружения дефектов в интерфейсах и во взаимодействии между интегрированными компонентами или системами.

Исследовательское тестирование (exploratory testing) - Неформальный метод проектирования тестов, при котором тестировщик активно контролирует проектирование тестов в то время, как эти тесты выполняются, и использует полученную во время тестирования информацию для проектирования новых и улучшенных тестов.

Итерация — однократное выполнение тела цикла.

**К**

Кеширование — технология, при которой документ загружается один раз и результат загрузки сохраняется. В следующий раз при обращении к документу, если он не был обновлен, то будет загружена сохраненная версия. Это помогает сократить время ожидания.

Консольное тестирование — тестирование приложений предназначенных для консолей.

Контекстное меню (Context menu) - элемент графического интерфейса операционной системы, представляющий собой список команд, вызываемый пользователем для выбора необходимого действия над выбранным объектом. Команды контекстного меню относятся к тому объекту, над которым это меню было вызвано.

Концепция дизайна (эскиз, макет, Mockup) – графический файл, который состоит из наиболее мелких картинок-слоев элементов общего рисунка.

Конфигурационное тестирование (англ. Configuration Testing) — специальный вид тестирования, направленный на проверку работы программного обеспечения при различных конфигурациях системы (заявленных платформах, поддерживаемых драйверах, при различных конфигурациях компьютеров и т.д.)

Крэш-лог (Crash Log) – файл, в котором хранится вся информация по ошибке неработоспособности/экстренного завершения работы программы.

**Л**

Логин (Login) – сетевое имя пользователя. Как часть реквизитов доступа к закрытым данным, логин почти всегда сопровождается паролем. У каждого пользователя может быть много логинов для доступа к разным данным. Они могут использоваться для доступа к системе управления сайтом, электронной почте и другим ресурсам.

Лэйаут — одна из составляющих графического дизайна; расположение визуальных элементов страницы или интерфейса, структурированное отображение информации на плоскости.

**М**

Матрица соответствия требований (англ. Traceability matrix) — это двумерная таблица, содержащая соответствие функциональных требований (functional requirements) продукта и подготовленных тестовых сценариев (test cases).

Метод проектирования тестов (test design technique) - Методика, используемая для создания и/или выбора тест кейсов

Мобильное тестирование — тестирование мобильных приложений.

Мобильное приложение (Mobile Application) – это специально разработанное приложение под конкретную мобильную платформу (iOS, Android, Windows Phone).

Мобильный web-сайт (Mobile Website) – специализированный сайт, адаптированный для просмотра и функционирования на мобильном устройстве.

Мобильная операционная система (mobile OS)– это операционная система для смартфонов, планшетов, КПК или других мобильных устройств.

Модульное тестирование (компонентное ) (component testing) — Тестирование отдельных компонентов программного обеспечения

Мультитач (Multitouch) – функция сенсорных систем ввода (сенсорный экран, сенсорная панель), осуществляющая одновременное определение координат двух и более точек касания

**Н**

Набор тестов (test suite, тест сьют): Комплект тестовых наборов для исследуемого компонента или системы, в котором обычно постусловие одного теста используется в качестве предусловия для последующего.

Нагрузочное тестирование — (load testing): Вид тестирования производительности, проводимый с целью оценить поведение компонента или системы под увеличивающейся нагрузкой (число одновременно работающих пользователей и/или число транзакций) для определения максимально допустимого уровня нагрузки для исследуемого компонента или системы.

Негативное тестирование (англ. Negative testing) — Тестирование, нацеленное на демонстрацию того, что система или компонент не работают. Негативное тестирование относится в большей степени к позиции тестировщика, нежели к определенному подходу к тестированию или метод проектирования тестов, например - тестирование с некорректными входными значениями или тестирование обработки исключений.

Нефункциональное тестирование — (non-functional testing): Тестирование атрибутов компонента или системы, не относящихся к функциональности, то есть надежность, эффективность, практичность, сопровождаемость и переносимость.

**О**

Обеспечение качества (Quality Assurance, QA) — совокупность мероприятий, охватывающих все технологические этапы разработки, выпуска и эксплуатации программного обеспечения

Отказ (failure) - Отклонение компонента или системы от ожидаемого выполнения, эксплуатации или результата.

Отладка (Дебаг) (англ.Debugging) — процесс, позволяющий получить программное обеспечение, функционирующее с требующимися характеристиками в заданной области входных данных.

Отчет о дефекте (defect/bug/problem report): Документ, содержащий отчет о любом недостатке в компоненте или системе, который может привести компонент или систему к невозможности выполнить требуемую функцию.

Ошибка (англ.Error) – действие, которое порождает неправильный результат.

**П**

Пароль (Password) – условное слово или набор знаков, предназначенный для подтверждения личности или полномочий. Пароли часто используются для защиты информации от несанкционированного доступа.

Парсер - Скрипт или программа, используемые для сбора информации с сайтов.

Повторное тестирование или подтверждающее тестирование (confirmation testing) - Тестирование, при котором выполняются тестовые сценарии, которые были не пройдены при последнем запуске, с целью подтвердить успешность исправлений.

Пользовательская история или юзерстори (user story): Высокоуровневое пользовательское или бизнес-требование, обычно использующееся в гибких методологиях разработки программного обеспечения. Обычно состоит из одного или нескольких предложений на разговорном или формальном языке, описывающих функциональность, необходимую пользователю, любые нефункциональные требования и включающих в себя критерии приемки.

Пользовательский опыт (user experience (UX) — совокупность впечатлений и субъективных ощущений от взаимодействия с интерфейсом программы или сайта.

Позитивное тестирование (англ. Positive testing) — тестирование ПО на то, как оно должно работать

Предугадывание ошибки (англ. Error Guessing — EG). : Метод проектирования тестов, когда опыт тестировщика используется для предугадывания того, какие дефекты могут быть в тестируемом компоненте или системе в результате сделанных ошибок, а также для разработки тестов специально для их выявления

Программное обеспечение (software): Компьютерные программы, алгоритмы и, зачастую, документация и данные, относящиеся к функционированию компьютерной системы.

Продакшн (production, прод/prod) — сервер (обычно подразумевают железо или как минимум виртуализированную операционную систему), на котором приложение выполняется для нужд конечных клиентов. Может существовать и тестовый сервер, на котором, в отличие от "боевого", приложение просто тестируют.

Приёмочное тестирование — (acceptance testing): Формальное тестирование по отношению к потребностям, требованиям и бизнес процессам пользователя, проводимое с целью определения соответствия системы критериям приёмки и дать возможность пользователям, заказчикам или иным авторизированым лицам определить, принимать систему или нет.

Причина / Следствие (англ. Cause/Effect — CE). Это, как правило, ввод комбинаций условий (причин), для получения ответа от системы (Следствие).

**Р**

Регрессионное тестирование (regression testing) — Тестирование уже протестированной программы, проводящееся после модификации для уверенности в том, что процесс модификации не внес или не активизировал ошибки в областях, не подвергавшихся изменениям. Проводится после изменений в коде программного продукта или его окружении.

Редирект — автоматическое перенаправление пользователей с одного сайта на другой или с одной страницы сайта на другую.

Релиз (release) — выпуск или выход в свет какого-либо продукта, компьютерной программы, а также окончательная версия программы.

Релиз-кандидат или RC (англ. Release candidate), Пре-релиз, иногда «гамма-версия» — стадия-кандидат на то, чтобы стать стабильной.

Релиз ноут или сопроводительная записка (release note) - Документ, идентифицирующий объекты для тестирования, их конфигурацию, текущий статус и полную необходимую информацию, предоставляемую разработчиками тестировщикам и иным заинтересованным лицам в начале этапа выполнения тестов.

Ручное тестирование (англ. Manual testing) — тестирование без использования программных средств, для проверки программы или сайта путём моделирования действий пользователя.

**С**

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

Сбой (англ.Failure) – несоответствие фактического результата работы компонента или системы ожидаемому результату.

Cвободное тестирование (ad hoc testing) - Тестирование, выполняемое неформально; без формальной подготовки тестов, формальных методов проектирования тестов, определения ожидаемых результатов и руководства по выполнению тестирования.

Серьезность ( Severity) — это атрибут, характеризующий влияние дефекта на работоспособность приложения.

Сенсорный экран (Touchscreen) – устройство ввода информации, представляющее собой экран, реагирующий на прикосновения к нему.

Сервер — в веб-разработке чаще всего под этим термином подразумевают набор серверного программного обеспечения, выполняющего сервисные (обслуживающие) задачи по запросу пользователя. Итогом работы сервера является предоставление пользователю доступа к определённым функциям или услугам.

Сессия (session) — механизм однозначной идентификации браузера, при котором на сервере создается файл для данного сеанса, в котором хранятся данные.

«Сначала мобильные» (mobile first) — подход к разработке адаптивных веб-сайтов, при котором проектирование и верстка интерфейса начинаются с версии для мобильных устройств.

Сниппет - Небольшой отрезок текстовой информации, который выводится рядом с ссылкой в поисковой выдаче. Другими словами, это краткое описание страницы сайта, релевантное поисковому запросу.

Событие в веб-аналитике — это действие пользователя на сайте, которое невозможно отследить в системах веб-аналитики с помощью задания url страниц, количества просмотров или длительность посещения.Например, событие позволяет отслеживать клики по кнопкам, которые не переводят пользователя на другой URL; скроллинг на странице; взаимодействие с чатом или баннером на сайте.Для отслеживания события необходимо прописать код события в том элементе, который необходимо отслеживать.

Страницы пагинации - ограниченный показ части информации, используемый для того, чтобы разбить большой массив данных на веб-странице и представить результаты по нескольким страницам. Пример: когда в категории товаров размещён список товаров, разбитый постранично для показа пользователю.

Симуляторы - программные и аппаратные средства, создающие впечатление действительности, отображая часть реальных явлений и свойств в виртуальной среде.

Смартфон (Smartphone) — мобильный телефон, дополненный функциональностью карманного персонального компьютера.

Спецификация — детальное описание того, как должно работать ПО.

Стадии разработки ПО — это этапы, которые проходят команды разработчиков ПО, прежде чем программа станет доступной для широко круга пользователей.

Сплит-тестирование, A/B testing, А/В тестирование, Split testing — метод маркетингового исследования, позволяющий сравнивать контрольную группу элементов с тестовой, в которой один или несколько показателей были изменены с целью выяснить, какие же из изменений могут улучшить целевой показатель. Пример A/B тестирования — исследование влияния размеров элементов интерфейса и их расположения на конверсию сайта.

Стрессовое тестирование (stress testing) - Вид тестирования производительности, оценивающий систему или компонент на граничных значениях рабочих нагрузок или за их пределами, или же в состоянии ограниченных ресурсов, таких как память или доступ к серверу.

Система отслеживания ошибок (англ. bug tracking system) — программа учета и/или контроля багов: Atlassian JIRA Bugzilla YouTrack Redmine etc.

Системное тестирование (system testing)— Процесс тестирования системы в целом с целью проверки того, что она соответствует установленным требованиям.

Статическое тестирование (англ.Static testing) — тестирование без запуска кода на исполнение.

Стандарт ISO/IEC/IEEE 29119 -

**Т**

Тестирование — процесс проверки соответствия заявленных к продукту требований и реально реализованной функциональности, осуществляемый путем наблюдения за его работой в искусственно созданных ситуациях и на ограниченном наборе тестов, выбранных определенным образом.

Тестирование удобства использования — тестирование, направленное на исследование того, насколько конечному пользователю понятно, как работать с продуктом, а также на то, насколько ему нравится использовать продукт.

Тестирование доступности (accessibility testing) - Тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты

Тестирование интерфейса (interface testing) - Тип интеграционного тестирования, связанный с тестированием интерфейсов между компонентами или системами.

Тестирование безопасности (safety testing) — тестирование, направленное на проверку способности приложения противостоять злонамеренным попыткам получения доступа к данным или функциям

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

Тестирование совместимости или тестирование взаимодействия (compatibility testing) — тестирование, направленное на проверку способности приложения работать в указанном окружении (браузер, мобильное ус-во и т.д.).

Тестирование использования ресурсов — (resource utilization testing): Процесс тестирования, исследующий использование ресурсов программным продуктом.

Тестирование восстанавливаемости (recoverability testing) Процесс тестирования, исследующий восстанавливаемость программного продукта.

Тестирование отказоустойчивости (failover testing) - Тестирование при помощи эмуляции отказов системы или реально вызываемых отказов в управляемом окружении. После вызванного отказа проверяется механизм отказоустойчивости с целью удостовериться, что данные не потеряны или не испорчены, и достигнут оговоренный уровень обслуживания (например, доступности функций или время отклика).

Тестирование производительности — (performance testing): Процесс тестирования с целью определить производительность программного продукта.

Тестирование сборки (англ. Build Verification Test) — тестирование направленное на определение соответствия, выпущенной версии, критериям качества для начала тестирования.

Тестирование пользовательского интерфейса (англ. UI Testing) — тестирование, выполняемое с целью определения, удобен ли некоторый искусственный объект (такой как веб-страница, пользовательский интерфейс или устройство) для его предполагаемого применения.

Тест кейс (англ. Test Case) — Набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определенной цели или тестового условия, таких как выполнения определенного пути программы или же для проверки соответствия определенному требованию.

Тест дизайн (англ. Test design) — это этап процесса тестирования ПО, на котором проектируются и создаются тестовые случаи (тест кейсы), в соответствии с

определёнными ранее критериями качества и целями тестирования

Тест план (англ. Test Plan) — Документ, описывающий цели, подходы, ресурсы и график запланированных тестовых активностей. Он определяет объекты тестирования, свойства для тестирования, задания, ответственных за задания, степень независимости каждого тестировщика, тестовое окружение, метод проектирования тестов, определяет используемые критерии входа и критерии выхода и причины их выбора, а также любые риски, требующие планирования на случай чрезвычайных обстоятельств

**У**

Утилита (англ. utility) — дополнительная компьютерная программа, входящая в состав общего программного обеспечения и предназначенная для специализированных задач, которые расширяют возможности операционных систем и оборудования.

**Ф**

Фавикон (FAVorites ICON, в переводе — «значок для избранного») - небольшое изображение отображается во вкладке браузера, в которой открыт сайт, рядом с title открытой страницы. Также его можно увидеть в результатах выдачи поисковой системы Яндекс слева от заголовка страницы.

Фокус-тест (англ. Focus test) — тестирование, проводимое с целью получения первичной реакции игроков. Необходимо для оценки удобства использования и того, как продукт принимается целевой аудиторией или сторонними людьми.

Фронтенд (frontend) область технологий для разработки веб-интерфейсов, включает вспомогательные инструменты и технологии, работающие в браузере. Внешняя часть серверной системы, отвечающая за приём данных, отправку ответов и генерацию кода для браузера.

Фидбек - отзыв пользователя.

Фронт-офис — отделы компании, которые в первую очередь напрямую работают с клиентами и для клиентов.

Функциональное тестирование (functional testing): - Тестирование, основанное на анализе спецификации функциональности компонента или системы.

Футер Footer (подвал) – область в самом конце страницы. Подвал располагается в самом низу и чаще всего (но не обязательно) дублирует пункты меню, логотип.

**Х**

Хлебные крошки (от англ. breadcrumbs) — дополнительная навигация на сайте, отображающая последовательность страниц, указывающих на местоположение пользователя на сайте относительно главной страницы.

Хостинг - Предоставление ресурсов для размещения информации (сайта) на сервере в сети интернет.

**Ч**

Чек-лист (англ. Check list) — это документ, описывающий что должно быть протестировано.

Черный ящик (англ. Black box) — Процедура создания и/или выбора тестовых сценариев, основанная на анализе функциональной или нефункциональной спецификации компонента или системы без знания внутренней структуры.

**Ш**

Шапка сайта (header) - Характерная повторяющаяся на всех страницах часть сайта, расположенная в самом верху страницы. Часто включает в себя логотип, навигацию, контакты.

**Э**

Эквивалентное Разделение (англ. Equivalence Partitioning — EP). Разработка тестов методом черного ящика, в которой тестовые сценарии создаются для проверки элементов эквивалентной области.

Экстеншн - маленькая программа, которая модифицирует и дополняет функциональность браузера

Эмулятор (emulator): Устройство, компьютерная программа или система, которая принимает те же самые входные данные и выдаёт те же самые выходные данные, что и данная система

Эквайринг — приём к оплате платежных карт через банк-эквайрер с помощью установленных на местах продаж платёжных терминалов. Интернет-экваринг позволяет совершить оплату картой через интернет с использованием веб-интерфейса, предоставляемого банком или платежной системой.

**Ю**

Юникод (Unicode) — принцип кодирования символов, позволяющий представить почти любой знак существующих письменных языков

[Вернуться к содержанию](https://github.com/netology-code/guides/blob/master/glossary_qa/README.md#%D1%81%D0%BE%D0%B4%D0%B5%D1%80%D0%B6%D0%B0%D0%BD%D0%B8%D0%B5)

**A**

ADB (Android Debug Bridge, отладочный мост Андроид) – компонент Android SDK, который устанавливает связь между устройством и компьютером и позволяет прямо на компьютере выполнять различные манипуляции с системой Android.

API (Application Programming Interface) — описание классов, методов, функций и переменных, которые приложение использует для выполнения поставленной задачи. Язык общения между программами.Ядро программы, в нем находятся все необходимые программе классы, методы, функции.

**В**

Breakpoints (переломные/контрольные точки) — изменение условий с одного на другое, при котором внешний вид сайта (раскладка) изменяется.

**С**

Completely Automatic Public Turing Test to Tell Computers and Humans Apart — полностью автоматический тест Тьюринга для различения компьютеров и людей — это задача, которую легко решает человек, но которую невозможно (или крайне трудно) научить решать компьютер. Применяются CAPTCHA для того, чтобы предотвратить множественные автоматические регистрации и отправления сообщений программами-роботами. Задача CAPTCHA — защита от спама, флуда и захвата аккаунтов.

Cookies - Создаваемые веб-страницей фрагменты данных, которые сохраняются после посещения того или иного сайта на компьютере пользователя в виде файлов. Всякий раз при открытии посещённых ранее страниц, веб-браузер пересылает эти файлы серверу. Cookies используются для аутентификации пользователей, хранения настроек и персональных данных, ведения статистики и др.

**I**

iTools – аналог iTunes, программа для просмотра и управления информацией на устройстве, снятия логов, установки билдов и снятия видео/скриншотов на базе iOS.

iTunes – программа для синхронизации устройств на базе iOS.

**K**

Kanban — система организации работы над проектами, в основе которой лежит принцип «точно в срок». Данная система способствует равномерному распределению нагрузки. Задачи по проекту заносятся в отдельный список по мере поступления. Выделяют несколько состояний задач: сделать, в работе, на проверке, завершены. В каждом состоянии есть ограниченное количество слотов. Пока слот не освободится, нельзя перенести задачу в заполненный список, что стимулирует не затягивать с выполнением задач. Весь процесс выполнения задач и ход работы должен быть прозрачен для каждого члена команды.

**L**

Landing Page, LP — целевая или посадочная страница— одностраничный сайт, который содержит описание продвигаемых продукта или услуги и стимулирует на целевое действие: покупку, регистрацию, звонок, заполнение анкеты.

**M**

Material Design — дизайн интерфейса приложений операционной системы Android. Создан для унификации всех продуктов Google. Основная идея заключается в использовании простых форм и взаимодействии между отдельными частями экрана.

**N**

NaN (Not A Number) — специальное значение, относится к числовому типу данных, возвращается когда математический оператор или функция не может вернуть результат вычисления.

**R**

Retina-экран — экран с таким количеством пикселей, при котором глаз человека на обычном для просмотра расстоянии уже не может различить отдельные точки, составляющие это изображение. Используется в устройствах Apple.

**S**

SDK (от англ. Software Development Kit) – набор разработчика программного обеспечения, который позволяет создавать приложения для определенного пакета, фрэймворка, игровой консоли, операционной системы или платформы.

Scrum — гибкая методология разработки. Основной акцент на качестве разработки. Процесс разработки делится на итерации, которые называются спринтами. По результатам спринта заказчику предоставляется рабочий продукт с наиболее приоритетными возможностями. Короткие промежутки времени (спринт длится 2-4 недели) делают процесс разработки гибким, позволяют быстро получать обратную связь от заказчика.

**X**

XPath (XML path) – язык запросов к элементам XML-документа, позволяет обращаться к частям документа. XPath призван реализовать навигацию по DOM в XML.

Xcode – интегрированная среда разработки программного обеспечения под OS X и iOS, разработанная корпорацией Apple.

**W**

Web-тестирование (Браузерное тестирование) — тестирование браузерных приложений.

**U**

UDID (Unique Device Identifier) – это уникальный идентификатор устройства, состоящий из 40 символов (для устройств: iPad, iPhone или iPod Touch).

UX (англ. User eXperience — опыт пользователя) — ощущение, испытываемое пользователем во время использования цифрового продукта.

UI (англ. User Interface — пользовательский интерфейс) — это инструмент, позволяющий осуществлять взаимодействие «пользователь — приложение>