Список терминов:

1. **Информационные технологии** – Информацио́нные техноло́гии (ИТ, также — информационно-коммуникационные технологии) — процессы, использующие совокупность средств и методов сбора, обработки, накопления и передачи данных (первичной информации) для получения информации нового качества о состоянии объекта, процесса, явления, информационного продукта, а также распространение информации и способы осуществления таких процессов и методов (ФЗ № 149-ФЗ).

Другие определения:

приёмы, способы и методы применения средств вычислительной техники при выполнении функций сбора, хранения, обработки, передачи и использования данных (ГОСТ 59853-2021);

ресурсы, необходимые для сбора, обработки, хранения и распространения информации (ISO/IEC 38500:2015).

1. **Системный интегратор** (факультативно) - Системный интегратор (англ. Systems integrator) — человек или компания-подрядчик, которые специализируются на объединении отдельных подсистем в единое целое. Также они решают задачи автоматизации
2. **Компьютерная сеть** - Компьютерная сеть образуется при физическом соединении (проводном или беспроводном) двух или более компьютеров для передачи данных между ними.

Это система, обеспечивающая обмен данными между вычислительными устройствами — компьютерами, серверами, маршрутизаторами и другим оборудованием или программным обеспечением. Для передачи информации могут быть использованы различные среды передачи данных. Помимо совокупности физических устройств и физических средств передачи данных, вычислительная сеть может быть оверлейной или виртуальной, то есть логически самостоятельной выделенной сетью, использующей ресурсы другой физической сети — вычислительной (например Интернет), телефонной сети, в том числе ТФОП и (или) среды передачи данных.

1. **Операционная система** - Операцио́нная систе́ма, сокр. ОС (англ. operating system, OS) — комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.
2. **Браузер** - Бра́узер, или веб-обозрева́тель (от англ. web browser, МФА) прикладное программное обеспечение для просмотра страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями; а также для решения других задач.
3. **Сервер (аппаратное обеспечение)** - Се́рвер (англ. server от англ. to serve — служить, мн. ч. се́рверы) — выделенный или специализированный компьютер для выполнения сервисного программного обеспечения (в том числе серверов тех или иных задач).
4. **Сервер (программное обеспечение)** - Се́рверное програ́ммное обеспечение (се́рвер, англ. server от to serve — служить; множественное число се́рверы, в разговорном языке также употребляется сервера́) — в информационных технологиях — программный компонент вычислительной системы, выполняющий сервисные (обслуживающие) функции по запросу клиента, предоставляя ему доступ к определённым ресурсам или услугам.
5. **ЦОД** - Центр (хранения и) обработки данных (ЦОД/ЦХОД) (от англ. data center) — это специализированное здание для размещения (хостинга) серверного и сетевого оборудования и подключения абонентов к каналам сети Интернет.
6. **Протокол передачи данных** - набор соглашений, позволяющий совершать обмен данными между различными компьютерами, сетями и программами. Именно протоколы определяют способ передачи сообщений, обработки ошибок в сети и позволяют разрабатывать стандарты, которые не были бы привязаны к одной определенной аппаратной платформе
7. **TCP/IP** - сетевая модель передачи данных, представленных в цифровом виде. Модель описывает способ передачи данных от источника информации к получателю. В модели предполагается прохождение информации через четыре уровня, каждый из которых описывается правилом (протоколом передачи). Наборы правил, решающих задачу по передаче данных, составляют стек протоколов передачи данных, на которых базируется Интернет[1][2]. Название TCP/IP происходит из двух важнейших протоколов семейства — Transmission Control Protocol (TCP) и Internet Protocol (IP), которые были первыми разработаны и описаны в данном стандарте.
8. **IP-адрес** - (от англ. Internet Protocol) — уникальный числовой идентификатор устройства в компьютерной сети, работающей по протоколу IP.
9. **URL -** Унифицированный указатель ресурса (от англ. Uniform Resource Locator, сокр. URL [ˌjuː ɑːr ˈel]) — система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса (файла). Используется как стандарт записи ссылок на объекты в Интернете (Гипертекстовые ссылки во «всемирной паутине» www).
10. **Программирование** - процесс создания компьютерных программ.
11. **HTML** - это язык разметки, который представляет простые правила оформления и компактный набор структурных и семантических элементов разметки (тегов). Это стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере.
12. **JavaScript** - это относительно простой объектно-ориентированный язык, предназначенный для создания небольших клиентских и серверных приложений для Internet.
13. **HTTP** - (англ. HyperText Transfer Protocol — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных. Основой HTTP является технология «клиент-сервер», то есть предполагается существование: Потребителей (клиентов), которые инициируют соединение и посылают запрос.
14. **HTTP cookie** - HTTP cookie (web cookie, cookie браузера) - это небольшой фрагмент данных, отправляемый сервером на браузер пользователя, который тот может сохранить и отсылать обратно с новым запросом к данному серверу.
15. **Java** - строго типизированный объектно-ориентированный язык программирования общего назначения,
16. **XML** - Extensible Markup Language (XML) - это язык разметки и формат файлов для хранения, передачи и восстановления произвольных данных. Он определяет набор правил кодирования документов в формате, который является как удобочитаемым, так и машиночитаемым.
17. **СУБД** - Система управления базами данных (СУБД) - специализированный комплекс программ, предназначенный для удобной и эффективной организации, контроля и администрирования баз данных.
18. **Отладка программы** (пример, для тестировщика – Firebug) - этап разработки компьютерной программы, на котором обнаруживают, локализуют и устраняют ошибки. Чтобы понять, где возникла ошибка, приходится: узнавать текущие значения переменных; выяснять, по какому пути выполнялась программа**.**
19. **Веб-интеграция (факультативно)** - процесс объединения при помощи Web-технологий различных бизнес-приложений и информационных систем компании в единую среду, которую обычно называют корпоративным информационным порталом (enterprise information portal — EIP).
20. **Веб-сервис** - (или веб-службы) идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами. Web-сервис (служба) – программа, которая организовывает взаимодействие между сайтами.
21. **Сервис-ориентированная архитектура** (факультативно) - (СОА, англ. service-oriented architecture- SOA) — модульный подход к разработке программного обеспечения, базирующийся на обеспечении удаленного по стандартизированным протоколам использования распределённых, слабо связанных[en] легко заменяемых компонентов (сервисов) со стандартизированными интерфейсами.
22. **Техническое задание** - документ или несколько документов, определяющих цель, структуру, свойства и методы какого-либо проекта, и исключающие двусмысленное толкование различными исполнителями.
23. **Бизнес-процесс** (факультативно) - совокупность взаимосвязанных мероприятий или работ, направленных на создание определённого продукта или услуги для потребителей.

**Андерклокинг** — снижение частоты работы оборудования.

**Баг (дефект)**— недостаток компонента или системы, который может привести к отказу определенной функциональности.

**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.

**Валидация**— определение соответствия разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе.

**Верификация**— процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа.

**Спецификация** — детальное описание того, как должно работать ПО.

**Система отслеживания ошибок** (*англ. bug tracking system*) — программа учета и/или контроля багов:

* Atlassian JIRA
* Bugzilla
* YouTrack
* Redmine
* etc.

**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.

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

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

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

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

Классификация по типу тестирования:  
**Мобильное тестирование** — тестирование мобильных приложений.  
**Консольное тестирование** — тестирование приложений предназначенных для консолей.  
**Web-тестирование** (Браузерное тестирование) — тестирование браузерных приложений.

Классификация по запуску кода на исполнение:  
**Статическое тестирование** (англ.*Static testing*) — тестирование без запуска кода на исполнение.  
**Динамическое тестирование** (*англ. Dynamic testing*) — тестирование с запуском кода на исполнение.

Классификация по доступу к коду и архитектуре ПО:  
**Черный ящик** (*англ. Black box*) — тестировщику не известно как устроена тестируемая система.  
**Белый ящик** (*англ. White box*) — тестировщику известно все детали реализации тестируемой системы.  
**Серый ящик** (*англ. Grey box*) — тестировщику известно только некоторые особенности устройства тестируемой системы.

Классификация по степени автоматизации:  
**Ручное тестирование** (*англ. Manual testing*) — тестирование ПО будучи его пользователем.  
**Автоматизированное тестирование** (*англ. Automated testing*) — тестирование ПО при помощи специальных программ.

Классификация по принципу работы с приложением:  
**Позитивное тестирование** (*англ. Positive testing*) — тестирование ПО на то, как оно должно работать.  
**Негативное тестирование** (*англ. Negative testing*) — тестирование ПО на то, как оно не должно работать.

Классификация по уровню детализации приложения:  
**Интеграционное тестирование** — тестирование взаимодействия и связей нескольких компонентов приложения.  
**Системное тестирование** — это тестирование всего приложения от начала и до конца.  
**Модульное тестирование** — тестирование на уровне отдельного функционального компонента приложения.

Классификация по целям и задачам:  
**Функциональное тестирование** — проверка корректности работы функциональности приложения.  
**Нефункциональное тестирование** — проверка нефункциональных особенностей приложения (удобство использования, совместимость, производительность, безопасность).  
**Инсталляционное тестирование** — проверка протекания стадии инсталляции (установки) приложения.  
**Регрессионное тестирование** — проверка на наличие багов, вызванных изменениями в приложении.  
**Повторное тестирование** — выполнение тест-кейсов, которые ранее обнаружили дефекты, с целью подтверждения устранения дефектов.  
**Приёмочное тестирование** — тестирование, направленное на проверку приложения с точки зрения конечного пользователя/заказчика  
**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** — тестирование, направленное на исследование пригодности продукта к использованию людьми с ограниченными возможностями  
**Тестирование интерфейса** — тестирование, направленное на проверку интерфейсов приложения или его компонентов.  
**Тестирование безопасности** — тестирование, направленное на проверку способности приложения противостоять злонамеренным попыткам получения доступа к данным или функциям  
**Тестирование интернационализации** — тестирование, направленное на проверку готовности продукта к работе с использованием различных языков и с учётом различных национальных и культурных особенностей.  
**Тестирование локализации** — тестирование, направленное на проверку корректности и качества адаптации продукта к использованию на том или ином языке с учётом национальных и культурных особенностей.  
**Тестирование совместимости** — тестирование, направленное на проверку способности приложения работать в указанном окружении (браузер, мобильное ус-во и т.д.).  
**Тестирование данных и баз данных** — тестирование, направленное на 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.  
**Сравнительное тестирование** — тестирование, направленное на сравнительный анализ преимуществ и недостатков разрабатываемого продукта по отношению к его основным конкурентам.  
**Демонстрационное тестирование** — формальный процесс демонстрации заказчику продукта с целью подтверждения, что продукт соответствует всем заявленным требованиям.  
**Избыточное 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.  
**Тестирование восстанавливаемости** — тестирование способности приложения восстанавливать свои функции и заданный уровень производительности, а также восстанавливать данные в случае возникновения критической ситуации.  
**Тестирование отказоустойчивости** — тестирование, заключающееся в эмуляции или реальном создании критических ситуаций с целью проверки способности приложения задействовать механизмы, предотвращающие нарушение работоспособности, производительности и повреждения данных.  
**Тестирование производительности** — исследование показателей скорости реакции приложения на внешние воздействия при различной по характеру и интенсивности нагрузке.  
**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.  
**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.)  
**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.

**Failure** — сбой (причём не обязательно аппаратный) в работе компонента, всей программы или системы.

**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».

**Анализ граничных значений** (*англ. Boundary Value Analysis — BVA*). Анализ граничных значений может быть применен к полям, записям, файлам, или к любого рода сущностям имеющим ограничения.

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

**Исследовательское (ad-hoc) тестирование** — это разработка и выполнения тестов в одно и то же время, что является противоположностью сценарного подхода.

**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.)

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

**Операционное тестирование** (*англ. Release Testing*). Даже если система удовлетворяет всем требованиям, важно убедиться в том, что она удовлетворяет нуждам пользователя и выполняет свою роль в среде своей эксплуатации, как это было определено в бизнес модели системы.

**Предугадывание ошибки** (*англ. Error Guessing — EG*). Это когда тест аналитик использует свои знания системы и способность к интерпретации спецификации на предмет того, чтобы «предугадать» при каких входных условиях система может выдать ошибку.

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

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

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

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

**Пре-альфа** (*англ. Pre-alpha*) — начальная стадия разработки. Период времени со старта разработки до выхода стадии Альфа. Также так называются программы, прошедшие стадию разработки, для первичной оценки функциональных возможностей в действии.

**Альфа-тестирование** (*англ. Alpha testing*) — имитация реальной работы с системой штатными разработчиками, либо реальная работа с системой потенциальными пользователями/заказчиком на ранней стадии разработки продукта, но в некоторых случаях может применяться для законченного продукта в качестве внутреннего приёмочного тестирования.

**Бета-тестирование** (*англ. Beta testing*) — интенсивное использование почти готовой версии продукта с целью выявления максимального числа ошибок в его работе для их последующего устранения перед окончательным выходом (релизом) продукта на рынок, к массовому потребителю.

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

**Релиз или RTM** (*англ. Release to manufacturing — промышленное издание*) — издание продукта, готового к тиражированию.

**Пост-релиз или Post-RTM** (*англ. Post-release to manufacturing*) — издание продукта, у которого есть несколько отличий от RTM и помечается как самая первая стадия разработки следующего продукта.

**Таблица принятия решений** (*англ. Decision table*) — инструмент для упорядочения сложных бизнес требований, которые должны быть реализованы в продукте.

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

**Тест-план** (*англ. Test Plan*) — это документ, описывающий весь объем работ по тестированию, а также оценки рисков с вариантами их разрешения.

**Тестирование взаимодействия** (*англ. Interoperability Testing*) — это функциональное тестирование, проверяющее способность приложения взаимодействовать с одним и более компонентами или системами.

**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.

**Тестовый случай** (*англ. Test Case*) — это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

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

**Эквивалентное Разделение** (*англ. Equivalence Partitioning — EP*). Как пример, у вас есть диапазон допустимых значений от 1 до 10, вы должны выбрать одно верное значение внутри интервала, скажем, 5, и одно неверное значение вне интервала — 0.

**Z-конфликт** (*англ. Z-fighting*) — наложение текстур друг на друга.

**Оверклокинг** (*англ. Overclocking*) — процесс увеличения частоты (и напряжения) компонента компьютера сверх штатных режимов с целью увеличения скорости его работы.