



**A/B testing (A/B-тестирование)** — метод маркетингового исследования, суть которого заключается в том, что контрольная группа элементов сравнивается с набором тестовых групп, в которых один или несколько показателей были изменены, для того, чтобы выяснить, какие из изменений улучшают целевой показатель.

Разновидностью A/B-тестирования является *многовариантное тестирование*. В этом случае тестируются не два целостных варианта, а сразу несколько элементов продукта или составных частей исследуемого объекта в различных сочетаниях, при которых каждый тестируемый элемент может быть двух видов (А или В).

**Acceptance criteria (критерии приемки)** — нужны нам для проверки каждой отдельной Истории Пользователя (фичи и т.п.) и подтверждения, что после реализации система работает, как этого хотел заказчик. Такие критерии будут почти уникальные для каждого элемента бэклога (Product Backlog Item) и должны уточняться, перед тем как команда возьмёт тот или иной элемент в итерацию (спринт).

**Acceptance testing (Приемочное тестирование)** — это тестирование ПО которое проводят для проверки на соответствие требованиям заказчика. Это финальный этап тестирования программы перед публичным запуском. Как только принято решение, что продукт полностью готов к использованию - приемочное тестирование должно подтвердить это. Часто приемочное тестирование означает привлечение команды независимых тестеров для анализа состояния продукта. Как правило, подобное тестирование выполняется, когда заказчик хочет узнать независимое мнение о качестве заказанного им продукта.

**Acquiring (получение)** — приём к оплате платёжных карт в качестве средства оплаты товара, работ, услуг.

**Adaptive layout (Адаптивная верстка)** — подход, предполагающий изменение дизайна в зависимости от поведения пользователя, размера



экрана, платформы и ориентации девайса. Другими словами, страница должна автоматически подстраиваться под разрешение, изменять размер картинок и т.д. Это позволит устранить нужду в разработке дизайна для каждого нового устройства, появляющегося в продаже.

**Add-on (Дополнение)** — добавление различного дополнительного материала к существующей программе или игре. Большинство аддонов добавляют новый функционал, расширяющий возможности основной версии.

**Agile software development (agile-методы)** - серия подходов к разработке программного обеспечения, ориентированных на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп, состоящих из специалистов различного профиля.

**AJAX** — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

**Alpha testing (Альфа-тестирование)** — имитация реальной работы с системой штатными разработчиками, или реальная работа с системой потенциальными пользователями/заказчиком. Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но в некоторых случаях может проводиться для готового продукта как внутреннее приемное тестирования.

**Amend** - (понятие, относящееся к Git) когда программист вносит изменения, не создавая новой версии — обычно когда разработчик ошибочно зафиксировал версию, но не залил её на сервер.



**API (Application Programming Interface)** - Интерфейс программирования приложений в виде набора готовых классов, функций, структур, констант и др., которые предоставляются приложением (библиотекой, сервисом) для использования во внешних программных продуктах.

**Application review (Обзор приложения)** – это сводка наиболее важной информации из всех отчетов аналитики для мобильных приложений. Он позволяет в целом оценивать эффективность приложения и отслеживать ключевые тенденции по основным группам отчетов: Конверсии, Аудитория, Поведение и Результативность.

**AR (Augmented reality - дополненная реальность)** — воспринимаемая смешанная реальность, создаваемая с использованием «дополненных» с помощью компьютера элементов воспринимаемой реальности (когда реальные объекты монтируются в поле восприятия).

**Array (Массив)** - тип или структура данных в виде набора компонентов (элементов массива), расположенных в памяти непосредственно друг за другом. При этом доступ к отдельным элементам массива осуществляется с помощью индексации, то есть через ссылку на массив с указанием номера (индекса) нужного элемента. Таким образом можно найти определенный элемент по его порядковому номеру - месту в цепочке данных. За счёт этого, в отличие от списка, массив является структурой данных, пригодной для осуществления произвольного доступа к её ячейкам.

**Asynchrony (Асинхронность)** - когда два события (создание события и обработка, например) - разделены во времени. Асинхронный часто употребляют в виде синонима “неблокирующий”.

**Authorization (Авторизация)** – это предоставленное какому-либо лицу право на совершение определенных действий в конкретной системе. Авторизация позволяет исключить доступ к информации третьих лиц.



Иногда для авторизации требуется предоставить значительное количество персональных данных, документов, пройти проверки и получить подтверждение.

**Automatic testing (Автоматизированное тестирование)** - часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения. Оно использует программные средства для выполнения тестов и проверки результатов выполнения, что помогает сократить время тестирования и упростить его процесс. Существует два основных подхода к автоматизации тестирования: тестирование на уровне кода и тестирование пользовательского интерфейса (имитация действий пользователя).

**BaaS (Backend as a Service «бэкенд как услуга»)** — модель, позволяющая разработчикам веб-приложений и мобильных приложений связать их приложения с серверным облачным хранилищем и API, выставляемыми серверными приложениями, а также предоставляющая такие функции, как: управление пользователями, извещающие уведомления, интеграция со службами социальных сетей. Поэтому другое название термина — **mBaaS**, мобильный бэкенд как услуга. Перечисленные выше услуги предоставляются посредством использования пользовательских комплектов для разработки программного обеспечения (SDK) и интерфейсов прикладного программирования (API).

**Back-end (Бекенд)** - область веб-технологий, работающих на сервере. Внутренняя часть серверной системы, которая занимается обработкой данных. Backend отвечает за логику работы программы или сайта.

**Backlog (Бэклог)** — это список всех работ проекта в Scrum. Можно сказать, что это ежедневник общего пользования.



**Beta testing (Бета-тестирование)** — интенсивное использование почти готовой версии продукта (как правило, программного или аппаратного обеспечения) с целью выявления максимального числа ошибок в его работе для их последующего устранения перед окончательным выходом (релизом) продукта на рынок, к массовому потребителю. В отличие от альфа-тестирования, проводимого силами штатных разработчиков или тестировщиков, бета-тестирование предполагает привлечение добровольцев из числа обычных будущих пользователей продукта, которым доступна упомянутая предварительная версия продукта.

**Big data** — совокупность подходов, инструментов и методов обработки структурированных и неструктурированных данных огромных объёмов и значительного многообразия для получения воспринимаемых человеком результатов, эффективных в условиях непрерывного прироста, распределения по многочисленным узлам вычислительной сети, сформировавшихся в конце 2000-х годов, альтернативных традиционным системам управления базами данных и решениям класса Business Intelligence.

**Black-box testing (Тестирование методом черного ящика)** - это когда тестировщик знает только то, что приложение должно делать. Но при этом он не может заглянуть внутрь ящика и посмотреть, как происходит преобразование исходных значений в итоговые.

**BLE (Bluetooth low energy emission - беспроводная технология Bluetooth с низким энергопотреблением)** — выпущенная в декабре 2009 года версия спецификации ядра беспроводной технологии Bluetooth, наиболее существенным достоинством которой является сверхмалое пиковое энергопотребление, среднее энергопотребление и энергопотребление в режиме простоя.

**Boolean** — логический тип данных, одной из особенностей которого, является то, что логические переменные могут принимать только 2



значения: True (правда) и False (ложь). присутствует в подавляющем большинстве языков программирования как самостоятельная сущность или реализуется через численный тип данных. В некоторых языках программирования за значение истина полагается 1, за значение ложь 0.

**Bootstrap** — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

**Branch (Ветвь в системах управления версиями)** — направление разработки, независимое от других. Ветвь представляет собой копию части хранилища (например, одного каталога), в которую можно вносить изменения, не влияющие на другие ветви. Документы в разных ветвях имеют одинаковую историю до точки ветвления и разные — после неё.

**Branch (Ветка)** - (понятие, относящееся к Git) направление разработки, независимое от других. Ветвь представляет собой копию части (как правило, одного каталога) хранилища, в которую можно вносить свои изменения, не влияющие на другие ветви. Документы в разных ветвях имеют одинаковую историю до точки ветвления и разные — после неё. Чтобы не работать с одним кодом большой командой программисты работают каждый в своей ветке, создавая их для определенных задач, а потом сливают в одну, пользуясь системой контроля версий. Ветка по умолчанию в Git'e называется master.

**Browser (Браузер)** – 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.

**Build (Билд)** - (часто используется в формулировке: “выкатить билд”). В



буквальном переводе — "сборка". Т.е. берём срез исходников (исходного кода и сопутствующих файлов), создаём некие исполняемые файлы. Получаем из исходников рабочие продукты. Они создаются как вручную по требованию, так и автоматическими системами сборок по расписанию.

**Business processes (Бизнес-процессы)** — это совокупность взаимосвязанных мероприятий или задач, направленных на создание определённого продукта или услуги для потребителей. В качестве графического описания деятельности применяются блок-схемы бизнес-процессов.

**Calability (Масштабируемость)** — означает способность системы, сети или процесса справляться с увеличением рабочей нагрузки (увеличивать свою производительность) при добавлении ресурсов (обычно аппаратных). Система называется масштабируемой, если она способна увеличивать производительность пропорционально дополнительным ресурсам. Масштабируемость можно оценить через отношение прироста производительности системы к приросту используемых ресурсов. Чем ближе это отношение к единице, тем лучше. Также под масштабируемостью понимается возможность наращивания дополнительных ресурсов без структурных изменений центрального узла системы.

**Check-out, Clone** — (понятие, относящееся к Git) извлечение документа из хранилища и создание рабочей копии.

**Class (Класс)** — тип данных, определяемый программистом. Класс представляет собой описание структуры объектов одного вида с набором методов их обработки. Аналогия с типом данных здесь напрашивается сама собой. Тип данных – это форма представления данных с набором операций. Отличие состоит в том, что тип данных либо уже определен в языке, либо формально составляется из уже существующих, а класс создается





программистом.

**Check-in, Commit, Submit** — (понятие, относящееся к Git) создание новой версии, фиксация изменений.

**Conflict (Конфликт)** — (понятие, относящееся к Git) ситуация, когда несколько пользователей сделали изменения одного и того же участка документа. Конфликт обнаруживается, когда один пользователь зафиксировал свои изменения (закомитил), а второй пытается зафиксировать и система сама не может корректно слить конфликтующие изменения. Поскольку программа может быть недостаточно разумна для того, чтобы определить, какое изменение является «корректным», второму пользователю нужно самому разрешить конфликт (resolve).

**Cron** — классический демон-планировщик задач в UNIX-подобных операционных системах, использующийся для периодического выполнения заданий в определённое время. Регулярные действия описываются инструкциями, помещёнными в файлы crontab и в специальные директории.

**Cross-platform solution (Кросс-платформенность)** — способность программного обеспечения работать более чем на одной аппаратной платформе и (или) операционной системе. Обеспечивается благодаря использованию высокоуровневых языков программирования, сред разработки и выполнения, поддерживающих условную компиляцию, компоновку и выполнение кода для различных платформ. Типичным примером является программное обеспечение, предназначенное для работы в операционных системах Linux и Windows одновременно.

**Code review** - инженерная практика в терминах гибкой методологии разработки. Это анализ (инспекция) кода с целью выявить ошибки, недочёты, расхождения в стиле написания кода, в соответствии написанного кода и поставленной задачи.





**Coding convention, Programming style (Стандарт оформления кода, стиль программирования)** - набор правил и соглашений, используемых при написании исходного кода на некотором языке программирования. Наличие общего стиля программирования облегчает понимание и поддержание исходного кода, написанного более чем одним программистом, а также упрощает взаимодействие нескольких человек при разработке программного обеспечения.

**Comment (Комментарий)** - пояснения к исходному тексту программы, находящиеся непосредственно внутри комментируемого кода. Синтаксис комментариев определяется языком программирования. С точки зрения компилятора или интерпретатора, комментарии — часть текста программы, не влияющая на её семантику. Комментарии не оказывают никакого влияния на результат компиляции программы или её интерпретацию. Помимо исходных текстов программ, комментарии также применяются в языках разметки и языках описания.

**Commit (Коммит)** - фиксация изменений или запись изменений в репозиторий. Коммит происходит на локальной машине.

**Compiler (Компилятор)** - программа, преобразующая текст, написанный на алгоритмическом языке, в программу, состоящую из машинных команд. Компилятор создает законченный вариант программы на машинном языке.

**Console (Консоль компьютера - пульт управления)** — совокупность устройств (в том числе устройств ввода-вывода), обеспечивающая взаимодействие человека-оператора с компьютером.



**Continuous Integration (CI- непрерывная интеграция)** — это практика разработки программного обеспечения, которая заключается в выполнении частых автоматизированных сборок проекта для скорейшего выявления и решения интеграционных проблем. В обычном проекте, где над разными частями системы разработчики трудятся независимо, стадия интеграции является заключительной. Она может непредсказуемо задержать окончание работ. Переход к непрерывной интеграции позволяет снизить трудоёмкость интеграции и сделать её более предсказуемой за счет наиболее раннего обнаружения и устранения ошибок и противоречий.

**Continuous Integration, CI** — это практика разработки программного обеспечения, которая заключается в слиянии рабочих копий в общую основную ветвь разработки несколько раз в день и выполнении частых автоматизированных сборок проекта для скорейшего выявления и решения интеграционных проблем. Также, этим термином могут называть программу, которая следит за вашим Source Control, и при появлении там изменений автоматически стягивает их, билдит, гоняет тесты (конечно, если их пишут) и так далее. В случае неудачи она дает об этом знать всем заинтересованным лицам, в первую очередь – последнему коммитеру.

**Credentials** - полномочия и учетные данные пользователя системы или самой программы при доступе к другой системе.

**CRM-система (Customer Relationship Management или Управление отношениями с клиентами)** — это прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.



**CRUD** (сокр. от англ. **create, read, update, delete**) — акроним, обозначающий четыре базовые функции, используемые при работе с персистентными хранилищами данных.

**CSS** — каскадные таблицы стилей. Формальный язык, с помощью которого задается внешний вид, оформление текстового документа, написанного при помощи html (языка разметки). Веб-сайты, как правило оформляются при помощи CSS - именно он отвечает за “красивости”.

**Database (База данных)** — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

**Database (База данных)** - совокупность связанных данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования. База данных независима от прикладных программ.

**DB2** — семейство систем управления реляционными базами данных, выпускаемых корпорацией IBM. Чаще всего, ссылаясь на DB2, имеют в виду реляционную систему управления базами данных DB2 Universal Database (DB2 UDB).

**Debugger (Отладчик)** — компьютерная программа, предназначенная для поиска ошибок в других программах, ядрах операционных систем, SQL-запросах и других видах кода. Отладчик позволяет выполнять трассировку, отслеживать, устанавливать или изменять значения переменных в процессе выполнения кода, устанавливать и удалять контрольные точки или условия остановки и т.д.



**Decompilation (Декомпиляция)** — процесс воссоздания исходного кода декомпилятором. Используется при обратной разработке программ.

**Definition of Done («Критерии готовности»)** — включают в себя ряд действий, которые нужно выполнить для того, чтобы сократить дополнительные действия между тем, когда команда говорит «мы сделали», и заказчик говорит «заверните, я беру». Если между этими двумя моментами вам нужно сделать ещё много телодвижений, то задумайтесь о том, каков же «Definition of Done» в вашей команде.

**Denormalization (Денормализация)** — намеренное приведение структуры базы данных в состояние, не соответствующее критериям нормализации, обычно проводимое с целью ускорения операций чтения из базы за счет добавления избыточных данных.

**Deploy (Деплой)** - (часто используется в формулировке “забилдить”, “задеплоить”). Процесс сбора исходного кода и перенос готового продукта в production, для использования непосредственными клиентами бизнеса. Если использовать аналогию: представьте, что вам, привезли тебе домой диван. Он полностью разобранный. Есть куча деталей - это исходный код. Первое, что нужно сделать это собрать диван — сбилдить. Вы собираете (билдите) диван и он стоит уже собранный. Потом диван надо отнести в гостинную и поставить там — то есть развернуть диван там, где вам нужна его функциональность. Это вы делаете деплой — несете и красиво ставите диван в середине комнаты.

**Deprecation (Неодобрение)** — в программировании некоторое указание нежелательности применения какой-либо части программы, процедуры или языка программирования. Чаще всего используется какой-либо стандартный для данного языка программирования или системы документации кода способ указания, например служебная метка (тег), специальная конструкция языка и т. п. В среде разработчиков компьютерных программ и документации, термин deprecation может



обозначать очередную стадию в жизненном цикле программного обеспечения, замену устаревших частей на новые.

**Development testing (Разработка через тестирование)** — техника разработки программного обеспечения, которая основывается на повторении очень коротких циклов разработки: сначала пишется тест, покрывающий желаемое изменение, затем пишется код, который позволит пройти тест, и под конец проводится рефакторинг нового кода к соответствующим стандартам.

**DevOps** (слияние англ. слов **Development** (разработка) и **Operations** (ИТ-операции)) – это новая методология разработки ПО. Она сосредоточена на коммуникации, сотрудничестве и интеграции между подразделениями разработки и эксплуатации. DevOps – это ответ на взаимную зависимость разработчиков и персонала ИТ-операций. Данная методология помогает организациям ускорить производство программных продуктов и предоставление своих сервисов.

**Discontinuity (Дискретность, Прерывность)** — определение того, что алгоритм должен представлять процесс решения задачи как последовательное выполнение простых (или ранее определенных) шагов. Каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего.

**DOM (Document Object Model)** — объектная модель, используемая для XML/HTML-документов. Согласно DOM-модели, документ является иерархией, деревом. Каждый HTML-тег образует узел дерева с типом «элемент». Вложенные в него теги становятся дочерними узлами. Для представления текста создаются узлы с типом «текст».

**Downtime** — время, в течение которого какое-либо устройство не может использоваться из-за сбоя в работе, произошедшим в нем самом или в



устройствах связанных с ним в единую систему.

**Duct tape (Костыль)** — (часто используется в формулировке: “костыли и велосипеды”). Быстрое "уродливое" решение проблемы, иначе требующей долгосрочного и ресурсоемкого исправления.

**Dynamic typing (Динамическая типизация)** - приём, широко используемый в языках программирования и языках спецификации, при котором переменная связывается с типом в момент присваивания значения, а не в момент объявления переменной. Таким образом, в различных участках программы одна и та же переменная может принимать значения разных типов. Примеры языков с динамической типизацией — Smalltalk, Python, Objective-C, Ruby, PHP, Perl, JavaScript, Lisp, xBase, Erlang.

**ECMAScript** — это встраиваемый расширяемый не имеющий средств ввода-вывода язык программирования, используемый в качестве основы для построения других скриптовых языков. Стандартизирован международной организацией ECMA в спецификации ECMA-262. Расширения языка, JavaScript, JScript и ActionScript, широко используются в вебе.

**E-commerce (Электронная коммерция)** — это сфера экономики, которая включает в себя все финансовые и торговые транзакции, осуществляемые при помощи компьютерных сетей, и бизнес-процессы, связанные с проведением таких транзакций.

**Electron** — система позволяющая создавать кроссплатформенные приложения используя одни только веб-технологии, такие как HTML, CSS и конечно, JS.

**Emulation (Эмуляция)** — комплекс программных, аппаратных средств или их сочетание, предназначенное для копирования (или эмулирования) функций одной вычислительной системы (гостя) на другой, отличной от



первой, вычислительной системе (хосте) таким образом, чтобы эмулированное поведение как можно ближе соответствовало поведению оригинальной системы (гостя). Целью является максимально точное воспроизведение поведения в отличие от разных форм компьютерного моделирования, в которых имитируется поведение некоторой абстрактной модели. Например, имитация игровой приставки (денди) на компьютере.

**Encapsulation (Инкапсуляция)** — это механизм контроля доступа. Она позволяет скрыть детали реализации, и открыть только то что необходимо в последующем использовании.

**Engine (Движок)** — (часто используется в формулировке: “написано на движке”) это центральная часть компьютерной программы, выполняющая основные функции этой программы.

**Enterprise (Энтерпрайс)** — система веб-технологий, обеспечивающих возможность оперативной совместной работы, обмена информацией и совместного принятия решений на предприятии.

**Environment (Среда программирования)**— интегрированная система разработки программ, в которой все программные средства, обеспечивающие разработку программ, имеют единый пользовательский интерфейс, общую базу данных и не требуют специального вызова.

**Epic (Эпик, Тема)** - это, по сути, большая user story (пользовательская история) — способ описания требований к разрабатываемой системе), которая может быть разбита на несколько небольших пользовательских историй. Тема, как правило, используются для описания крупных и сложных элементов из бэклога продукта (список приоритезированных запросов) с низким приоритетом, еще не описанных детально.

**Extreme Programming, XP (Экстремальное программирование)** - это упрощенная методология организации разработки программ для





небольших и средних по размеру команд разработчиков, занимающихся созданием программного продукта в условиях неясных или быстро меняющихся требований.

**Flow (Поток)** — это по сути последовательность инструкций, которые выполняются параллельно с другими потоками. Каждая программа создает по меньшей мере один поток: основной, который запускает функцию `main()`. Программа, использующая только главный поток, является однопоточной; если добавить один или более потоков, она станет многопоточной.

**Follow-up letter (Письмо о дальнейших действиях)** - письмо, которое направляется деловому контрагенту после первоначальной встречи или любой другой встречи для закрепления договоренностей и намеченных шагов.

**Fork** - копия репозитория. Его также можно рассматривать как внешнюю ветку для текущего репозитория. Копия вашего открытого репозитория на Гитхабе может быть сделана любым пользователем, после чего он может прислать изменения в ваш репозиторий через пулреквест.

**Fragmentation (Фрагментация платформы)** — ситуация, когда у какой-то вычислительной платформы становится настолько много моделей аппаратуры и версий ОС, что становится практически невозможным написать программу, хорошо работающую на всех устройствах, созданных на базе данной вычислительной платформы. Рассмотрим на примере платформы Android - на ней выпущено огромное количество девайсов, от разных компаний, с различным разрешением и размером экранов и при создании ПО нужно все это учитывать.

**Frameworks (Фрейм-ворки)** – это про-грамм-ные про-дук-ты, кото-рые упро-щают созда-ние и под-держку тех-ни-че-ски слож-ных или



нагру-жен-ных про-ек-тов. Фрейм-ворк, как пра-ви-ло, содер-жит только базо-вые про-грамм-ные моду-ли, а все спе-ци-фич-ные для про-екта ком-по-ненты реа-ли-зу-ются раз-ра-бот-чи-ком на их осно-ве. Тем самым дости-га-ется не только высо-кая ско-рость раз-ра-бот-ки, но и боль-шая про-из-во-ди-тель-ность и надёж-ность решений.

**Front-end (Фрондэнд)** - область технологий для разработки веб-интерфейсов, включает вспомогательные инструменты и технологии, работающие в браузере. Внешняя часть серверной системы, отвечающая за прием данных, отправку ответов и генерацию кода для браузера. Чаще всего это то, видят пользователи.

**Fullstack developer (фулстек)** - это веб разработчик имеющий знания и опыт в разработке как frontend так и backend составляющих. Обычно требуется в компании где нет деления на frontend/backend разработчиков, когда компания маленькая или фронтенд/бекенд составляющая на проекте не требует большого количества разработчиков. Можно сказать, что это специалист, который обладает знаниями для разработки веб проекта от А до Я, пусть и не очень глубокими.

**Git (Гит)** — это распределенная система для управления версиями файлов.

**GitHub** — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub. Git дает возможность обновлять и править файлы проекта с учетом изменений внесенных другими.

**Guideline (гайдлайн, руководство)** — технический документ, содержащий набор стандартов по использованию логотипа, товарных знаков, персонажей бренда, а также варианты их масштабирование, цветового и шрифтового исполнение, правильного расположения на различных носителях.



**Hardware (Аппаратное обеспечение)** — электронные и механические части вычислительного устройства, входящие в состав системы или сети, исключая программное обеспечение и данные. Аппаратное обеспечение включает: компьютеры и логические устройства, внешние устройства и диагностическую аппаратуру, энергетическое оборудование, батареи и аккумуляторы.

**Hash (Хеш-сумма, хеш, хеш-код)** — результат обработки неких данных хеш-функцией. Значение хеш-суммы может использоваться для проверки целостности данных, их идентификации и поиска (например в P2P-сетях), а также заменять собой данные, которые небезопасно хранить в явном виде (например, пароли, ответы на вопросы тестов и т. д.). Также алгоритмы хеширования используются для проверки целостности и подлинности файлов.

**Hashing (Хеширование)** — преобразование массива входных данных произвольной длины в (выходную) битовую строку фиксированной длины, выполняемое определённым алгоритмом. Функция, реализующая алгоритм и выполняющая преобразование, называется «хеш-функцией» или «функцией свёртки». Исходные данные называются входным массивом, «ключом» или «сообщением». Результат преобразования (выходные данные) называется «хешем», «хеш-кодом», «хеш-суммой», «сводкой сообщения».

**Heatmap (Тепловая карта)** — это графическое представление данных, где дополнительные переменные отображаются при помощи цвета.

**High-level programming language (Высокоуровневый язык программирования)** — язык программирования, разработанный для быстроты и удобства использования программистом. Такой тип языков используется повсеместно и является самым востребованным из всех. Значительно проще синтаксис, абстракции упрощают жизнь, большие



стандартные библиотеки (не надо изобретать велосипед), в целом все просто и интуитивно понятно. Минус в сравнительно низкой производительность, кроме того страдает архитектура (изменения могут вноситься в любом месте кода, а не там, где это действительно нужно).

**Highload** — крайне относительное понятие. Оно никогда не измеряется количеством запросов или скоростью работы сайта, т.к. попросту нет такого понятия, как "средний сайт". Все сайты специфичны и одинаковое количество запросов может приводить к совершенно разным нагрузкам на разные ресурсы. Если Ваш сайт не справляется с нагрузкой — все, теперь Вы в клубе highload.

**Highload system (Высоконагруженные системы)** — это, по большому счету, те же веб-сайты, только с очень большой аудиторией, и как следствие с большой нагрузкой, требующей оптимизированной серверной части веб-сайта. Качественной характеристикой для высоконагруженной системы является пропускная способность этой системы, она описывает количество работы, которую должна уметь выполнять система в единицу времени. Примерами высоконагруженных систем могут служить такие веб-сайты как Google, Facebook, Вконтакте, Flickr, Amazon, YouTube, Wikimedia, LiveJournal, Twitter, MySpace и многие другие. Также, к высоконагруженным системам можно отнести браузерные многопользовательские игры.

**HTML** - язык разметки гипертекста, то есть документов в интернете. Все веб-страницы в сети создаются при помощи html-разметки. Современные браузеры преобразуют HTML в форматированный текст и отображают его на разных устройствах.

**HTTP (HyperText Transfer Protocol)** — протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов в формате «HTML», в настоящий момент используется для передачи произвольных данных). Основой HTTP является технология «клиент-



сервер», то есть предполагается существование: потребителей (клиентов), которые инициируют соединение и посылают запрос; поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно.

**In-App Purchase** - сервис покупки виртуальных товаров внутри приложения (например, игровой валюты, новых уровней, игровых предметов и т.д.). Применяется он в основном в играх, в тех случаях, когда встает вопрос о необходимости заработка на своем творении, а распространять его платно не особо хочется (или нет смысла).

**Infrastructure (Инфраструктура)** — комплекс взаимосвязанных обслуживающих структур или объектов, составляющих и/или обеспечивающих основу, обеспечивающую функционирование системы.

**Inheritance (Наследование)** — это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым, родительским или суперклассом. Новый класс — потомком, наследником или производным классом

**Instance (Экземпляр класса)** — это описание конкретного объекта в памяти. Класс описывает свойства и методы, которые будут доступны у объекта, построенного по описанию, заложенному в классе. Экземпляры используются для представления (моделирования) конкретных сущностей реального мира.

**Integrated development environment, IDE (Интегрированная среда разработки, ИСР)** — комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).



**Interface (Интерфейс)** – это визуальное средство, позволяющее пользователю взаимодействовать с программой или операционной системой устройства, облегчающее понимание того, как необходимо пользоваться этим устройством.

**Ionic framework** — один из самых широко обсуждаемых фреймворков. Как гласит официальный сайт, Ionic — это SDK для создания гибридных мобильных приложений, набор CSS и JS компонент, созданный на основе AngularJS, SASS, Apache Cordova.

**IoT (Internet of Things, Интернет вещей)** — методология вычислительной сети физических предметов («вещей»), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой, рассматривающая организацию таких сетей как явление, способное перестроить экономические и общественные процессы, исключающее из части действий и операций необходимость участия человека.

**JavaScript, JS** – это язык программирования, с помощью которого веб-страницам придается интерактивность. С его помощью создаются приложения, которые включаются в HTML-код (например, анкеты или формы регистрации, которые заполняются пользователем).

**JSON** - простой, основанный на использовании текста, способ хранить и передавать структурированные данные. С помощью простого синтаксиса можно легко хранить все, что угодно, начиная от одного числа до строк, массивов и объектов, в простом тексте. Также можно связывать между собой массивы и объекты, создавая сложные структуры данных.

**Library (Библиотека)** — сборник подпрограмм или объектов, используемых для разработки программного обеспечения (ПО).

**Lint** — статический анализатор для языка программирования Си, который сообщал о подозрительных или непереносимых на другие платформы



выражениях. В начале XXI века термин стал нарицательным для всех программ такого типа. Как инструмент программа лишь анализирует статический исходный код, не скомпилированный в отличие от дебаггеров.

**Linting (Линтинг, Линт)** — Статический анализ кода. Проверка синтаксиса и корректности (неинициализированные переменные, например), не запуская компиляцию или выполнение (в зависимости от языка). Помогает сразу исправить ошибки, связанные с “стилем написания кода”.

**Load balancer (балансировка нагрузки, или выравнивание нагрузки)** — метод распределения заданий между несколькими сетевыми устройствами (например, серверами) с целью оптимизации использования ресурсов, сокращения времени обслуживания запросов, горизонтального масштабирования кластера (динамическое добавление/удаление устройств), а также обеспечения отказоустойчивости (резервирования).

**Load testing (Нагрузочное тестирование)** — подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).

**Localization (Локализация)** — перевод пользовательского интерфейса, документации и сопутствующих файлов программного обеспечения с одного языка на другой. Локализация не ограничивается переводом интерфейса на другой язык. Это многоуровневая операция, первые шаги которой можно выполнить, даже оставив интерфейс нетронутым, а последние — требуют взаимодействия программистов, дизайнеров и переводчиков, например, чтобы в компьютерной игре переводить даже прорисованные элементы.





**Low-level programming language (Низкоуровневый язык программирования)** — язык программирования, близкий к программированию непосредственно в машинных кодах используемого реального или виртуального процессора. В таких языках необходимо управлять выделением памяти и строго за этим следить.

**Machine-learning (Машинное обучение)** — класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решению множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.

**Manual testing (Ручное тестирование)** - (часто используется формулировка “мануальное тестирование”) это процесс поиска дефектов в работе программы, когда тестировщик проверяет работоспособность всех компонентов программы, как если бы он был пользователем. Часто, для точности проверки, тестировщик использует заранее заготовленный план тестирования, в котором отмечены наиболее важные аспекты работы программы. В роли тестировщиков могут выступать и обычные пользователи, сообщая разработчикам о найденных ошибках.

**Master** - ветка по умолчанию в Git. Главная ветка в которой содержится основная рабочая версия кода.

**Merchant account (Мерчант аккаунт)** – это один из видов банковского счета, с помощью которого в Интернете осуществляются платежи по кредитным или дебетовым пластиковым картам. Именно через Мерчант-счета производится зачисление средств на конкретный банковский счет интернет-предпринимателя.



**Merge** - операция слияния двух и более веток.

**Merge, Integration (Мёрдж, Слияние)** - (понятие, относящееся к Git) объединение независимых изменений в единую версию документа. Осуществляется, когда два человека изменили один и тот же файл или при переносе изменений из одной ветки в другую.

**Method (Метод)** – функция, работающая с объектом. В данном случае функция - это фрагмент программного кода, к которому можно обратиться из другого места программы. В большинстве случаев с функцией связывается идентификатор, но многие языки допускают и безымянные функции.

**Microservice architecture (Микросервисная архитектура)** — это подход к созданию приложения, подразумевающий отказ от единой, монолитной структуры. То есть вместо того чтобы исполнять все ограниченные контексты приложения на сервере с помощью внутрипроцессных взаимодействий, мы используем несколько небольших приложений, каждое из которых соответствует какому-то ограниченному контексту. Причём эти приложения работают на разных серверах и взаимодействуют друг с другом по сети, например посредством HTTP.

**Microservices (Микросервисы)** — современное представление сервис-ориентированной архитектуры (SOA), используемое для создания распределенных программных систем. Как и в SOA, модули в архитектуре микросервисов взаимодействуют по сети друг с другом для выполнения цели. Преимуществом распределения различных функций системы в различные небольшие модули является то, что это усиливает связность и уменьшает связанность.

**Mid-level programming languages (Среднеуровневые языки)** - часто используются для первых шагов в «мир программирования», так как они довольно несложны в изучении, но уже могут использоваться для каких-



либо реальных проектов. Памятью все еще можно управлять, но не обязательно; богатые стандартные библиотеки; компилируются в байт-код, чем упрощается взаимодействие с другими языками.

**Minutes, Meeting notes (Протокол)** - это письменный документ, отражающий ход митинга. Это может быть стенограмма, или просто запись встречи, из которой потом будут сформированы фолоапы.

**Minification (Минификация)** в языках программирования и особенно в JavaScript — процесс, направленный на уменьшение размера исходного кода путём удаления ненужных символов без изменения его функциональности. Исходный код сокращен является особенно полезным для программ на интерпретируемых языках, развернутые и передаются в интернете (например, JavaScript), потому что это уменьшает объем данных, который должен быть передан.

**Mock-Up (Мокап)** - неработающая модель, выполненная в натуральную величину и выглядящая так, как будет выглядеть работающий экземпляр. То есть, сделанная в фотопрограмме веб-страница, отданная на верстку — это мокап. А дизайном она станет, когда появится интерактивность. Иногда, мокапы могут быть относительно динамичными и интерактивными.

**MongoDB** — документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++. Вместо хранения данных в таблицах, состоящих из отдельных строк, как в реляционных базах, MongoDB сохраняет данные в коллекциях, состоящих из документов. Документ — это большой JSON объект без заранее определенного формата и схемы.

**Monolithic architecture (Монолитная архитектура)** - При монолитной архитектуре программа или сервис не имеет какой-либо явно выраженной



внутренней структуры. Это просто набор процедур, использующих общие глобальные данные, и вызываемые друг другом или пользователем.

**MS SQL (Microsoft SQL Server)** — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основным используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

**Multithreading (Многопоточность)** — свойство платформы (например, операционной системы, виртуальной машины и т. д.) или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков, выполняющихся «параллельно», то есть без предписанного порядка во времени. При выполнении некоторых задач такое разделение может достичь более эффективного использования ресурсов вычислительной машины.

**Native advertising (Нативная реклама)** – это «естественная» реклама: она приобретает форму и характеристики той платформы, на которой размещается. Она не похожа на обычную рекламу, поэтому, по идее, не должна вызывать чувство отторжения у читателей, а также помогает избежать явления баннерной слепоты.

**NDA (Non-disclosure agreement)** — соглашение о неразглашении конфиденциальной информации. Такой контракт можно подписывать с любым субъектом, которому Вы доверяете доступ к конфиденциальной информации.

**Neural network (Искусственная нейронная сеть, ИНС)** — математическая модель, а также её программное или аппаратное



воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

**NFC (Near Field Communication)** — это технология беспроводной высокочастотной связи малого радиуса действия (до 10 см), позволяющая осуществлять бесконтактный обмен данными между устройствами, расположенными на небольших расстояниях: например, между считывающим терминалом и сотовым телефоном или пластиковой смарт-картой.

**NPM** — это пакетный менеджер node.js. С его помощью можно управлять модулями и зависимостями.

**Obfuscate (Обфусцировать, Обфускация)** - запутывание кода. Приведение исходного текста или исполняемого кода программы к виду, сохраняющему её функциональность, но затрудняющему анализ, понимание алгоритмов работы и модификацию при декомпиляции.

**Object (Объект)** – это отдельный представитель класса, имеющий конкретное состояние и поведение, полностью определяемое классом.

**Object-oriented programming (Объектно-ориентированное программирование)** — парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

**Open-source software (Открытое программное обеспечение)** — программное обеспечение с открытым исходным кодом. Исходный код таких программ доступен для просмотра, изучения и изменения, что позволяет пользователю принять участие в доработке самой открытой программы, использовать код для создания новых программ и исправления в них ошибок — через заимствование исходного кода, если это позволяет совместимость лицензий, или через изучение использованных алгоритмов,



структур данных, технологий, методик и интерфейсов (поскольку исходный код может существенно дополнять документацию, а при отсутствии таковой сам служит документацией).

**ORM (англ. Object-Relational Mapping, рус. объектно-реляционное отображение)** — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных». Существуют как проприетарные, так и свободные реализации этой технологии.

**Package manager (Система управления пакетами)** — набор программного обеспечения, позволяющего управлять процессом установки, удаления, настройки и обновления различных компонентов программного обеспечения. Системы управления пакетами активно используются в различных дистрибутивах операционной системы Linux и других UNIX-подобных операционных системах. Программное обеспечение представляется в виде особых пакетов, содержащих помимо дистрибутива программного обеспечения набор определённых метаданных, которые могут включать в себя полное имя пакета, номер версии, описание пакета, имя разработчика, контрольную сумму, отношения с другими пакетами. Метаданные сохраняются в системной базе данных пакетов.

**Packet (Пакет)** — это определённым образом оформленный блок данных, передаваемый по сети в пакетном режиме. Компьютерные линии связи, которые не поддерживают пакетный режим, как, например, традиционная телекоммуникационная связь точка-точка, передают данные просто в виде последовательности байтов, символов или битов поодиночке. Если данные сформированы в пакеты, битрейт коммуникационной среды можно более эффективно распределить между пользователями, чем в сети с коммутацией каналов. При использовании сетей с коммутацией пакетов можно надёжно гарантировать пороговый битрейт, ниже которого он



опускаться

не

будет.

**Parsing (Парсинг)** - программа (скрипт), дающая возможность компьютеру «читать» – сравнивать предложенные слова с имеющимися во Всемирной сети, называется парсером. Сфера применения таких программ очень широка, но все они работают практически по одному алгоритму. Парсер - то же, что граббер, т. е. скрипт или программа, которые используются для сбора информации с сайтов для последующего размещения на собственных ресурсах. Первоначально под парсингом подразумевался процесс поиска определенной информации в большом фрагменте текста, а так же разбиение данных на смысловые части.

**PCI DSS (Payment Card Industry Data Security Standard)** — стандарт безопасности данных индустрии платёжных карт? представляет собой совокупность 12 детализированных требований по обеспечению безопасности данных о держателях платёжных карт, которые передаются, хранятся и обрабатываются в информационных инфраструктурах организаций. Принятие соответствующих мер по обеспечению соответствия требованиям стандарта подразумевает комплексный подход к обеспечению информационной безопасности данных платёжных карт.

**PII (Personally identifiable information Лично идентифицируемая информация)** — используемая в законах о защите информации и конфиденциальности, представляет собой информацию, которая может использоваться сама по себе или с другой информацией для идентификации контакта.

**Placeholder (Плэйсхолдер)** или иными словами картинка-заглушка, означает прямоугольное изображение определённого цвета и размеров для временного использования вместо какого-то другого, возможно ещё не имеющегося в наличии изображения.





**Platform as a Service (PaaS, «платформа как услуга»)** — модель предоставления облачных вычислений, при которой потребитель получает доступ к использованию информационно-технологических платформ: операционных систем, систем управления базами данных, связующему программному обеспечению, средствам разработки и тестирования, размещенным у облачного провайдера. В этой модели вся информационно-технологическая инфраструктура, включая вычислительные сети, серверы, системы хранения, целиком управляется провайдером, провайдером же определяется набор доступных для потребителей видов платформ и набор управляемых параметров платформ, а потребителю предоставляется возможность использовать платформы, создавать их виртуальные экземпляры, устанавливать, разрабатывать, тестировать, эксплуатировать на них прикладное программное обеспечение, при этом динамически изменяя количество потребляемых вычислительных ресурсов.

**Plugin (Плагин)** — независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Плагины обычно выполняются в виде библиотек общего пользования.

**Polymorphism (Полиморфизм)** — это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

**Port (порт)** — натуральное число, записываемое в заголовках протоколов транспортного уровня модели OSI (TCP, UDP, SCTP, DCCP). Используется для определения процесса-получателя пакета в пределах одного хоста.

Ошибка 503 - Service Temporarily Unavailable

Ошибка 502 - Bad Gateway

Ошибка 503 - Service Temporarily Unavailable

Ошибка 500 - Внутренняя ошибка сервера



**PostgreSQL (Пост-Грес-Кью-Эл)** — свободная объектно-реляционная система управления базами данных (СУБД). Существует в реализациях для множества UNIX-подобных платформ, включая AIX, различные BSD-системы, HP-UX, IRIX, Linux, macOS, Solaris/OpenSolaris, Tru64, QNX, а также для Microsoft Windows.

**Production (Продакшн)** — (часто используется в формулировке: “выкатить в продакшн”). Имеется ввиду production server. Это рабочее, так называемое "боевое" окружение, в котором производится работа с реальными клиентами и актуальными данными. Production-сервер - это сервер, относящийся к production-окружению. На production-сервере должна находиться стабильная, оттестированная сборка.

**Programming language (Язык программирования)** — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые она будет выполнять.

**Prototype (Прототип)** - модель для тестирования концепции или процесса. Могут быть вставлены картинки, появиться цвето-тональные градации и т.д. Иногда, прототипом называют разные типы визуализации и макетирования. В этом случае прототипом будут называть и мокап и блок-схему и даже скетч.

**Pull request** — (понятие, относящееся к Git) это отправка ваших изменений в репозитории на проверку другим программистам или тимлиду. Пример ситуации. Вы нашли ошибку в каком-то фреймворке или библиотеке и хотите исправить ошибку и отправить разработчикам, чтобы они внесли ваши изменения в master-ветку репозитория. Pull request это отправка ваших изменений, но без вливания кода сразу в репозиторий. Вначале ваш commit проверят другие разработчики.



**Pull, Update** - (понятие, относящееся к Git) запрос для получения новых версий из хранилища.

**Push** - (понятие, относящееся к Git) залить новую версию в хранилище.

**PWA (progressive web app)** — это название группы приложений, которые используют стек Web технологий (JS + HTML + CSS) и позволяют соединить простоту использования Web сайта со специфичными для нативных приложений операционной системы UX и техническими возможностями. Основное предназначение PWA увеличить конверсию, количество пользователей и удобство использования Web приложений на мобильных устройствах.

**P2P (Peer to peer – равный к равному) сеть** — это децентрализованная компьютерная сеть, основанная на равноправии участников. Часто в такой сети отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и выполняет функции сервера. В отличие от архитектуры клиент-сервера, такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов. Участниками сети являются пиры.

**QA (Управление качеством)** - (от английского Quality Assurance) это измерение и управление качеством процесса, который используется для создания качественного продукта. Отличается от Quality Control, тем, что Quality Control - это поддержка качества продукта в текущий момент времени. Quality Control ориентирован на продукт, разрабатываемый в данный момент.

**RDP (Remote Desktop Protocol, Протокол удалённого рабочего стола)** — проприетарный протокол прикладного уровня, позаимствованный Microsoft из купленной у PictureTel (ныне известной как Polycom) телекоммуникационной программы Liveshare Plus (названной впоследствии NetMeeting), использующийся для обеспечения удалённой



работы пользователя с сервером, на котором запущен сервис терминальных подключений. Клиенты существуют практически для всех версий Windows (включая Windows CE, Phone и Mobile), Linux, FreeBSD, Mac OS X, iOS, Android, Symbian. По умолчанию используется порт TCP 3389.

**Recursion (Рекурсия)** — это свойство объекта подражать самому себе. Объект является рекурсивным если его части выглядят также как весь объект.

**Refactoring (Рефакторинг)** - переработка кода. Комплекс мер, направленных на увеличение производительности, уменьшение количества кода и улучшение его читабельности. Процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и ставящий целью облегчить понимание её работы. В основе рефакторинга лежит последовательность небольших эквивалентных (то есть сохраняющих поведение) преобразований. Поскольку каждое преобразование маленькое, программисту легче проследить за его правильностью, и в то же время вся последовательность может привести к существенной перестройке программы и улучшению её согласованности и чёткости.

**Register (Регистр)** - специальная ячейка памяти (последовательное или параллельное логическое устройство), с функцией кратковременного хранения числа или команды и выполнения над ними некоторых операций. Отличие ее от обычной ячейки памяти в том, что она способна преобразовывать двоичный код а не только его хранить.

**Regular Expressions (Регулярные выражения)** - (иногда употребляется в виде сленговых сокращений: регэкспы или регексы) это система обработки текста, основанная на специальной системе записи образцов для поиска. Образец - паттерн, задающий правило поиска, по-русски иногда называют «шаблоном», «маской».



**Regular expressions (Регулярные выражения)** — формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов. Для поиска используется строка-образец, состоящая из символов и метасимволов и задающая правило поиска. Для манипуляций с текстом дополнительно задается строка замены, которая также может содержать в себе специальные символы.

**Relational database (Реляционная база данных)** - представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного вида. Каждая строка таблицы содержит данные об одном объекте (например, автомобиле, компьютере, клиенте), а столбцы таблицы содержат различные характеристики этих объектов - атрибуты (например, номер двигателя, марка процессора, телефоны фирм или клиентов).

**Release (Релиз)** - выпуск готового для использования продукта (программы / кода / библиотеки). Обычно он содержит все обновления, исправления и является версией, готовой для использования конечным потребителем.

**Release notes (Замечания к версии программного продукта)** — часть документации программного обеспечения, в которой описываются изменения между выпускаемой и предыдущей версиями этого ПО. Могут составляться для внешних пользователей, для отдела тестирования, документирования, маркетинга.

**Repository (Репозиторий)** - (понятие, относящееся к Git) хранилище документов — место, где система управления версиями хранит все документы вместе с историей их изменения и другой служебной информацией.

**Repository (Репозиторий, хранилище)** — место, где хранятся и поддерживаются какие-либо данные. Чаще всего данные в репозитории хранятся в виде файлов, доступных для дальнейшего распространения по



сети.

**REST (Representational State Transfer)** — архитектурный стиль взаимодействия компонентов распределенного приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределенной гипермедиа-системы. В определенных случаях (интернет-магазины, поисковые системы, прочие системы, основанные на данных) это приводит к повышению производительности и упрощению архитектуры. В широком смысле компоненты в REST взаимодействуют наподобие взаимодействия клиентов и серверов во Всемирной паутине. REST является альтернативой RPC.

**Retention (Удержание пользователей)** — одна из ключевых метрик для любого продукта, а в особенности для мобильных игр. Основная цель — сделать так, чтобы пользователь продолжал играть спустя несколько недель и даже месяцев.

**Reverse engineering, Back engineering (Обратная разработка)** - исследование некоторого готового устройства, в нашем случае программы, а также документации на нее с целью понять принцип ее работы. Например, чтобы обнаружить недокументированные возможности (в том числе программные закладки), сделать изменение или воспроизвести программу или иной объект с аналогичными функциями, но без прямого копирования.

**RFP (от английского request for proposal)** — это заявка на оказание услуги или создание проекта, которую создает заказчик для проведения конкурса. В ней находят отражение те цели, которых хочет достичь заказчик, KPI, критерии к участникам тендера и ряд других важных показателей.

**ROI (returning of investments)** — финансовый коэффициент, иллюстрирующий уровень доходности или убыточности бизнеса, учитывая сумму сделанных в этот бизнес инвестиций. ROI обычно



выражается в процентах, реже - в виде дроби. Показатель ROI является отношением суммы прибыли или убытков к сумме инвестиций.

**Routing (Маршрутизация)** — процесс определения маршрута следования информации в сетях связи. Маршруты могут задаваться административно (статические маршруты), либо вычисляться с помощью алгоритмов маршрутизации, базируясь на информации о топологии и состоянии сети, полученной с помощью протоколов маршрутизации (динамические маршруты).

**Rubber layout (Резиновая вёрстка)** — Блоки меняют свою ширину в зависимости от размера окна браузера. Она может принимать максимальное и минимальное значение (свойство max-width). Но нельзя по мере уменьшения экрана из 50% сделать 100%.

**SaaS (англ. *software as a service* — программное обеспечение как услуга)** — одна из форм облачных вычислений, модель обслуживания, при которой подписчикам предоставляется готовое прикладное программное обеспечение, полностью обслуживаемое провайдером. Поставщик в этой модели самостоятельно управляет приложением, предоставляя заказчикам доступ к функциям с клиентских устройств, как правило через мобильное приложение или веб-браузер. Основное преимущество модели SaaS для потребителя услуги состоит в отсутствии затрат, связанных с установкой, обновлением и поддержкой работоспособности оборудования и работающего на нём программного обеспечения.

**Sandbox (Песочница)** — специально выделенная среда для безопасного исполнения компьютерных программ. Обычно представляет собой жёстко контролируемый набор ресурсов для исполнения гостевой программы — например, место на диске или в памяти. Доступ к сети, возможность общаться с главной операционной системой или считывать информацию с устройств ввода обычно либо частично эмулируют, либо сильно





ограничивают. Песочницы представляют собой пример виртуализации.

**Scalability (Масштабируемость)** — способность системы, сети или процесса справиться с увеличением рабочей нагрузки (увеличивать свою производительность) при добавлении ресурсов (обычно аппаратных). Система называется масштабируемой, если она способна увеличивать производительность пропорционально дополнительным ресурсам.

**Screencast (Скринкаст)** — цифровая видеозапись информации, выводимой на экран компьютера, также известная как video screen capture (досл. «видеозахват экрана»). Часто сопровождается голосовыми комментариями.

**Script (Скрипт)** — (от английского “сценарий”) это программа или программный файл, которые автоматизируют некоторую задачу, которую пользователь делал бы вместо этого вручную. Скрипты создают определенный необходимый функционал. Часто скриптом называется исполняемая процедура, которая выполняется со стороны сервера вследствие запроса, отправленного с определенной интернет-страницы.

**Scrum (Скрам)** — это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые спринтами (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет. Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всём его протяжении. При этом строго фиксированная небольшая длительность спринта придаёт процессу разработки предсказуемость и гибкость.

**SDK (Software Development Kit)** — комплект средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного



обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

**SDLC (Software development lifecycle - жизненный цикл разработки системы)** - это серия из шести основных фаз, через которые проходит любая программная система.

- 1) Планирование системы
- 2) Анализ системы
- 3) Дизайн системы
- 4) Разработка, внедрение и развертывание
- 5) Опытная эксплуатация и интеграция
- 6) Поддержка системы

**Server (Сервер)** - это компьютер, выделенный из группы персональных компьютеров (или рабочих станций) для выполнения какой-либо сервисной задачи без непосредственного участия человека.

**Server application (Серверное приложение)** – это специализированная программа, которая принимает запросы клиентов, обрабатывает их и дает ответы на эти вопросы.

**Simulation (Симулякр, Симуляция, Полнофункциональный прототип)** - на сложных проектах тоже модель для тестирования концепции или процесса. Это модель уровня Hi-Fi (high fidelity — высокой точности, в отличие от прототипа, который является моделью низкой точности — Lo-Fi). Для создания симуляций обычно используется программа iRise. Там используются библиотеки визуалов, позволяющие изобразить страницы очень близко к конечному виду, есть выпадающие списки, кнопки, меняющие свой вид при наведении, навигация между экранами, и т. д.

**Signed application (Сертификаты для подписи кода)** — используются разработчиками на всех платформах, для цифрового подписывания



приложения и программ, распространяемых через Интернет. Подписанный код, указывающий имя издателя и защищая программу от инъекций вредоносного кода и других видов повреждений, обеспечивает такую же безопасность, как и у купленного в магазине программного обеспечения.

**Single Page Application, SPA (Приложение одной страницы)** – это web-приложение, размещенное на одной web-странице, которая для обеспечения работы загружает весь необходимый код вместе с загрузкой самой страницы. В этом случае пользователь может взаимодействовать с интерфейсом без постоянных перезагрузок страницы так, как будто это десктопное приложение. Приложения такого типа появились сравнительно недавно, с началом эры HTML5. Хорошим примером SPA можно назвать почту Gmail или Twitter.

**Sketch (Набросок, Эскиз)** - первоначальный моментальный набросок от руки того, что пришло в голову.

**Software architecture (Архитектура)** - это общее устройство кода приложения: библиотеки, модули, классы, функции и их отношения. Т. е. общее описание. А программисты пишут конкретный код в этой архитектуре. Качественная архитектура приложения должна быть расширяемой. т. е. позволять добавлять новые части.

**Spinner** – это выпадающий список, позволяющий выбрать одно значение. Он позволяет сэкономить место на экране.

**Sprint (Спринт)** - отрезок времени в scrum, который берется для выполнения определенного (ограниченного) списка задач. Рекомендуется брать 2-4 недели (длительность определяется командой один раз).



**SQL (Structured query language)** — формальный непроцедурный язык программирования, применяемый для создания, модификации и управления данными.

**SSL-сертификат** – это своего рода уникальная цифровая подпись вашего сайта. Такой сертификат нужен, в первую очередь, банкам, платежным системам и другим организациям, работающим с персональными данными, – для защиты транзакций и предотвращения несанкционированного доступа к информации.

**Stack (Стек)** — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»).

**Stack Overflow** — популярная система вопросов и ответов о программировании, разработанная Джоэлем Спольски и Джеффом Этвудом в 2008 году. Является частью Stack Exchange Network. Как и в других системах подобного рода, Stack Overflow предоставляет возможность оценивать вопросы и ответы, что поднимает или понижает репутацию зарегистрированных пользователей (вариант игрофикации). Проект создан с использованием ASP.NET 4 и ASP.NET MVC.

**Static typing (Статическая типизация)** - означает, что все проверки типов данных выполняются на этапе компиляции, а не на этапе выполнения программы. Переменная, параметр подпрограммы, возвращаемое значение функции связывается с типом в момент объявления и тип не может быть изменён позже (переменная или параметр будут принимать, а функция — возвращать значения только этого типа). Примеры языков с такой типизацией: Ада, C++, D, Java, ML, Паскаль.

**String (Строковый тип)** — тип данных, значениями которого является произвольная последовательность (строка) символов алфавита. Каждая переменная такого типа (строковая переменная) может быть представлена



фиксированным количеством байтов либо иметь произвольную длину.

**Style guide (Руководство по стилю)** — это набор стандартов для написания и оформления программного обеспечения для общего использования или для конкретной версии. Объясняет как конкретно должен использоваться каждый дизайн-элемент.

**Syntactic sugar (Синтаксический сахар)** — это синтаксические возможности, применение которых не влияет на поведение программы, но делает использование языка более удобным для человека.

**User session (Сессия пользователя)** — это время, которое пользователь провел в мобильном приложении с момента запуска до окончания его использования.

Эта метрика может свидетельствовать о том, насколько интересно пользователю проводить время в приложении. То есть это косвенный критерий качества. Кроме того, если в вашем приложении есть платный контент, но с увеличением средней продолжительности сессии вырастает и вероятность того, что пользователь решит заплатить. В большинстве проектов платящие пользователи проводят в приложении больше времени, чем неплатящие.

**VR (virtual reality - виртуальная реальность)** — созданный техническими средствами мир (объекты и субъекты), передаваемый человеку через его ощущения: зрение, слух, обоняние, осязание и другие. Виртуальная реальность имитирует как воздействие, так и реакции на воздействие. Для создания убедительного комплекса ощущений реальности компьютерный синтез свойств и реакций виртуальной реальности производится в реальном времени.

**Tag (Тег)** — элемент языка разметки гипертекста. Более правильное название — дескриптор. Например, текст Википедии, заключенный между начальным тегом и конечным тегом “small” предписывает отображать его



меньше основного текста. В XML тег является элементом документа, а текст, содержащийся между начальным и конечным тегом — содержанием элемента.

**Test Plan (План тестирования, Тест План)** - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

**Test Run (Тесты для запуска)** - это комбинация тест скриптов или тестовых наборов для последующего совместного запуска (последовательного или параллельного, в зависимости от преследуемых целей и возможностей инструмента для автоматизированного тестирования).

**Test-driven development (TDD - Разработка через тестирование)** — является передовой техникой разработки программного обеспечения, основывается на повторении кратких циклов разработки: изначально пишется тест, который покрывает желаемое изменение, а далее пишется сам код, позволяющий пройти данный тест, и в конце проводится рефакторинг нового кода к соответствующим стандартам.

**Tester (Тестировщик)** - специалист, занимающийся тестированием. В его обязанность входит поиск вероятных ошибок и сбоев в функционировании объекта тестирования (продукта, программы, и тд). Тестировщик выполняет задачи по подготовке и проведению тестирования системы в соответствии с установленными в компании регламентами.

**Third party data (Данные третьих сторон)** - любые внешние данные, их источником могут быть сервисы email-рассылок, платёжные системы, сторонние сайты, сервисы обработки и хранения данных DMP.

**Ticket** - это виртуальная карточка с описанием ошибки.



**Token (Токен)** - замещение объекта неким обозначением/идентификатором (часто просто целым числом или строкой). Например: идентификатор сессии это классический токен. То есть вместо персональных данных пользователя и времени посещения сайта мы пишем просто номер (код). Токены широко применяются в системах авторизации/идентификации и прочее. Технически обычно реализуются в виде записи в базе данных, где токен является идентификатором записи о данных юзера или предоставленного доступа.

**Transliteration (Транслитерация)** — точная передача знаков одной письменности знаками другой письменности, при которой каждый знак (или последовательность знаков) одной системы письма передаётся одним и тем же знаком (или последовательностью знаков) другой системы письма.

**Tutorial (Тьюториал, учебное пособие)** - Это пошаговое (поэтапное) объяснение какого-либо процесса, инструкция по выполнению чего-либо. Тьюториалы бывают: текстовые, графические и видео. Популярны графические тьюториалы для художников, где наглядно расписано что и как рисовать. Существуют видео-тьюториалы, например, тьюториалы которые учат геймеров проходить туры в видео-играх, или видео-инструкции по пользованию определенным программным обеспечением.

**UI, User Interface (Пользовательский интерфейс)** – элементы внешнего вида системы (кнопки, иконки, меню, рубрики), цель которого – максимально эффективно и удобно организовать взаимодействие с пользователем.

**Unit testing (Модульное тестирование, или юнит-тестирование)** — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы. Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное



изменение кода к регрессии, то есть к появлению ошибок в уже протестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

**Unity** — это инструмент для разработки двух- и трёхмерных приложений и игр, работающий под операционными системами Windows, Linux и OS X.

**Uptime** — это время непрерывной работы сайта.

**Usability (Юзабилити)** - удобство использования интерфейса с точки зрения пользователя. Позитивные впечатления при использовании программного обеспечения, логичность и простота в расположении элементов управления.

**Use Case (Сценарий использования)** - это описание поведения системы, когда она взаимодействует с кем-то (или чем-то) из внешней среды. Система может отвечать на внешние запросы, а может сама выступать инициатором взаимодействия. Другими словами, сценарий использования описывает, «кто» и «что» может сделать с рассматриваемой системой, или что система может сделать с «кем» или «чем». Методика сценариев использования применяется для выявления требований к поведению системы, известных также как пользовательские и функциональные требования.

**User Flow (Путь пользователя)** - переход пользователей от одного сценария взаимодействия с интерфейсом к другому, пользовательский маршрут. Task flows, или потоки задач — этапы, через которые проходит пользователь при навигации по вашему ресурсу или приложению для достижения определенных целей. Они отличаются тем, что не зависят от особенностей конкретного посетителя, например, от точки вхождения на ресурс, клиентского статуса, количества действий, а описывают работу системы в целом, давая представление обо всех возможностях продукта.





**User Story (Пользовательские истории)** — способ описания требований к разрабатываемой системе, сформулированных как одно или более предложений на повседневном или деловом языке пользователя. Пользовательские истории используются гибкими методологиями разработки программного обеспечения для спецификации требований (вместе с приемочными испытаниями). Каждая пользовательская история ограничена в размере и сложности.

**UX, User Experience (Опыт взаимодействия, Пользовательский опыт)** - характеризует личное восприятие человеком функциональных и эмоциональных характеристик продукта или услуги в процессе использования. Опыт пользователя, в том числе, связан с таким понятием как юзабилити, применяемым при разработке и анализе пользовательских интерфейсов приложений. Разработчики уделяют значительное внимание изучению и проектированию опыта пользователя на всех этапах создания продукта, начиная с самого раннего — этапа планирования, — до проведения его тестирования.

**Variable (Переменная)** — это область памяти, имеющая имя, которое иначе называют идентификатором. Давая переменной имя, программист одновременно тем же именем называет и область памяти, куда будут записываться значения переменной для хранения.

**Version Control System (VCS) (Система управления версиями)** — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

**Version Control System, VCS (Система контроля версий)** — это система, записывающая изменения в файл или набор файлов в течение времени.



Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

**Virtual machine (Виртуальная машина)** — программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы (целевой, или гостевой платформы) и исполняющая программы для target-платформы на host-платформе (host — хост-платформа, платформа-хозяин) или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы; также спецификация некоторой вычислительной среды.

**VPN (Virtual Private Network, Виртуальная частная сеть)** — обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений (логическую сеть) поверх другой сети (например, Интернет). Несмотря на то, что коммуникации осуществляются по сетям с меньшим или неизвестным уровнем доверия (например, по публичным сетям), уровень доверия к построенной логической сети не зависит от уровня доверия к базовым сетям благодаря использованию средств криптографии (шифрования, аутентификации, инфраструктуры открытых ключей, средств для защиты от повторов и изменений, передаваемых по логической сети сообщений).

**Web Application (Веб-приложение)** — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами.



**Web layout (Вёрстка)** — создание структуры html-кода, размещающего элементы веб-страницы (изображения, текст и т. д.) в окне браузера, согласно разработанному макету, таким образом, чтобы элементы дизайна выглядели аналогично макету.

**WebSocket** — протокол полнодуплексной связи (может передавать и принимать одновременно) поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени. В настоящее время в W3C осуществляется стандартизация API Web Sockets.

**White-box testing (Тестирование методом белого ящика)** - это когда тестировщик имеет доступ к исходному коду приложения и может исследовать его для получения подсказок, как следует тестировать, какие значения вводить. Он может заглянуть внутрь белого ящика и посмотреть, как происходит преобразование исходных значений в итоговые.

**Wireframe (Блок-схема)** - схема или чертеж, представляющие «скелет» страницы сайта или приложения. Никаких украшений, только расположение и примерные размеры заголовков, текстовых блоков, иллюстраций, мультимедиа- и навигационных панелей. Нужен, в первую очередь, чтобы определить структуру.

**Workflow** — это система обеспечения выполнения задач, поставленных перед исполнителями в рамках процессного управления.

**XML** - это расширяемый язык разметки (Extensible Markup Language), разработанный специально для размещения информации в World Wide Web, наряду с HTML, который давно стал стандартным языком создания Web-страниц. С помощью XML можно создавать текст и размечать его при помощи обрамляющих тегов, превращая каждое слово, предложение или фрагмент в идентифицируемую, сортируемую информацию.



**Аргумент функции** в программировании — значение, передаваемое функции, или его символьное имя.

**Верстка сайта** — это процесс создания веб-страницы по готовому макету. Обычно в качестве макета используется графический шаблон, созданный в Adobe Photoshop, он называется PSD (PhotoShop Document) шаблоном. Процесс верстки включает в себя создание кода страницы при помощи понятного браузерам языка разметки гипертекста (HTML), и оформление её с помощью каскадных таблиц стилей (CSS).

**Говнокод** — это применение не самых лучших (с точки зрения большинства) решений проблемы. Ну к примеру говнокодом назовут выход из цикла установкой счетчика в максимальное значение. Это совершенно рабочее решение, но для решения такой проблемы есть уже проверенные практики и специальные методы. Говоря другими словами, Quality Assurance гарантирует, что процесс поставлен правильно и дает предсказуемый результат, в то время как Quality Control гарантирует, что продукт удовлетворяет указанному набору требований. На рынке СНГ понятие QA часто используют, чтобы называть так тестировщиков, в большей степени тех, кто занимается автоматическим тестированием.

**Комментарии** — пояснения к исходному тексту программы, находящиеся непосредственно внутри комментируемого кода. Синтаксис комментариев определяется языком программирования. С точки зрения компилятора или интерпретатора, комментарии — часть текста программы, не влияющая на её семантику. Комментарии не оказывают никакого влияния на результат компиляции программы или её интерпретацию. Помимо исходных текстов программ, комментарии также применяются в языках разметки и языках описания.

**Контекстная реклама** — тип интернет-рекламы, при котором рекламное объявление показывается в соответствии с содержанием, контекстом интернет-страницы. Контекстная реклама действует избирательно и



отображается посетителям интернет-страницы, сфера интересов которых потенциально совпадает/пересекается с тематикой рекламируемого товара либо услуги, целевой аудитории, что повышает вероятность их отклика на рекламу.

**Лог** - файл с записями о событиях в хронологическом порядке. Различают регистрацию внешних событий и протоколирование работы самой программы - источника записей (хотя часто всё записывается в единый файл). Это что-то наподобие подробного отчёта. дневника. Туда программа записывает какое действие и во сколько она выполнила.

**Локальный репозиторий** — репозиторий, расположенный на локальном компьютере разработчика в каталоге. Именно в нём происходит разработка и фиксация изменений, которые отправляются на удалённый репозиторий.

**Модуль** — функционально законченный фрагмент программы. Во многих языках (но далеко не обязательно) оформляется в виде отдельного файла с исходным кодом или поименованной непрерывной её части. Некоторые языки предусматривают объединение модулей в пакеты.

**Педалить код** — заниматься программированием в проекте на позиции, требующей больших объемов кода и минимума творчества.

**Протокол передачи данных** — стандарт, описывающий правила взаимодействия функциональных блоков при передаче данных.

**Псевдокод** — компактный (зачастую неформальный) язык описания алгоритмов, использующий ключевые слова императивных языков программирования, но опускающий несущественные подробности и специфический синтаксис. Псевдокод обычно опускает детали, несущественные для понимания алгоритма человеком. Такими несущественными деталями могут быть описания переменных, системно-зависимый код и подпрограммы. Главная цель использования псевдокода



— обеспечить понимание алгоритма человеком, сделать описание более воспринимаемым, чем исходный код на языке программирования. Псевдокод широко используется в учебниках и научно-технических публикациях, а также на начальных стадиях разработки компьютерных программ.

**Реляционная база данных** – это совокупность отношений, содержащих всю информацию, которая должна храниться в БД. Однако пользователи могут воспринимать такую базу данных как совокупность таблиц.

**Служба теневого копирования тома** (англ. Volume Shadow Copy Service, VSS) — служба операционной системы Windows, позволяющая копировать файлы, с которыми в данный момент времени ведётся работа, и даже с системными и заблокированными файлами. Служба необходима для работы следующих программ: восстановление системы, программ архивации и др.

**Сущность (Объект в программировании)** — некоторая сущность в компьютерном пространстве, обладающая определённым состоянием и поведением, имеющая заданные значения свойств (атрибутов) и операций над ними (методов)<sup>[1]</sup>. Как правило, при рассмотрении объектов выделяется то, что объекты принадлежат одному или нескольким классам, которые определяют поведение (являются моделью) объекта.

**Технический долг** — это метафора-неологизм, обозначающая плохую продуманность структуры системы, не продуманную архитектуру программного обеспечения или некачественную разработку ПО.

**Удалённый репозиторий** — репозиторий, находящийся на удалённом сервере. Это общий репозиторий, в который приходят все изменения и из которого забираются все обновления.