



IT словарь

Язык IT-шников достаточно специфичный, используется очень много англоязычных слов, некоторые в совершенно непонятной форме. Поэтому мы подготовили для вас список слов, которые встречаются на курсах SkillUP (курс «Тестирование с нуля», основной курс «Тестирование ПО»).

1. **Ад-хок (ad hoc)** - тестирование, похожее на разведовательное, но считается, что вы уже работали с тестируемой программой до этого и знаете, как она *работает*. Делая этот тест, вы за короткий срок должны поработать с программой, обхватив главные аспекты её работы, и задокументировать ее странное поведение. Как правило, сценарий для такого вида тестирования на 100% импровизированный.
2. **Альфа тестирование (alfa testing)** - 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 проводится для готового продукта в качестве внутреннего приёмного тестирования.
3. **Анекспектед бехевиер (unexpected behavior)** - неожиданное поведение.
4. **Апдейт (update)** — обновление.
5. **Апдейтнутый (от...up to date)** - в актуальном состоянии.
6. **Апликация (application)** - программа. Другое название — дополнение.
7. **Артефакт (artifact)** — артефакт.
8. **Аутпут (Output)** - исходящие данные, результат.
9. **Аутсорсинг (Outsourcing)** - передача компанией или части её задач или процессов сторонним исполнителям на условиях субподряда. Сделка, по которой работа делается людьми с внешней компании, которая обычно также является экспертом в этом виде работ. Аутсорсинг часто используется для сокращения затрат.
10. **Баг (bug)** - ошибка в программном коде.
11. **Баг трекер (bug tracker)** - прикладная программа, которая помогает разработчикам программного обеспечения (программистам, тестировщикам и т.д.) учитывать и

контролировать ошибки (баги), найденные в программах, вопросы о функциональности, решений и обновлений, пожелания пользователей, а также следить за процессом устранения ошибок и выполнение или невыполнение пожеланий.

12. **Багзилла (bugzilla)** - система отслеживания ошибок и ведения задач (англ. Bugtracker tool) с веб-оболочкой.

13. **Баундри вельюс (boundary values)** - предельные значения, граничные значения.

14. **Бейсд (based)** - основан (на основе чего-то).

15. **Бек лог (backlog)** - это документ, который содержит список требований к функциональности, которые упорядочены согласно степени важности. Backlog демонстрирует перечень того, что должно быть реализовано.

16. **Бета тестирование (beta testing)** - 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 .

17. **Билд (build)** - дословно: строить, конструировать. В языке разработки ПО - объединение отдельных модулей программы в единую рабочую систему. Также употребляется для определения номера построения программы с определенного кода (изменения в коде означают, что следующий билд будет иметь другую версию).

18. **Блайнд спот (blind spot)** - дословно: слепое пятно.

19. **Блэк бокс (black box)** - Тестирование "черного ящика". При тестировании «черного ящика» рассматриваются системные характеристики программ, игнорируется их внутренняя логическая структура. Исчерпывающее тестирование, как правило, невозможно.

20. **Вайт бокс (white box)** - тестирование "белого ящика". Обычно тестирование "белого ящика" основано на анализе управляющей структуры программы. Программа считается полностью проверенной, если проведено исчерпывающее тестирование маршрутов (путей) ее графа управления.

21. **Валидный (valid)** - настоящий, действующий, правильный.

22. **Верификация (verification)** - проверка.

23. **Верифайд (verified)** - проверенный.

24. **Виснуть, зависать** - состояние ПО, когда невозможно выполнить какое либо действие. Программа не отвечает на запросы и не выполняет никаких заметных действий.

25. **Вотерфол (waterfall)** - дословно: водопад. Последовательный метод разработки программного обеспечения, названный так из-за диаграммы, похожей на водопад.

26. **Гайдлайн (guideline)** - инструкция, руководство. Часто выпускаются одними разработчиками для других, чтобы гарантировать совместимость отдельных модулей

программного обеспечения и других его элементов (в том числе графического дизайна, использования интерфейсов и т.д.) с разработками сторонних компаний.

27. **Генерить (generate)** - сгенерировать.

28. **Голд плейтинг (gold plating)** - дословно: золочение. В языке тестирования - то, что не имеет полезного действия.

29. **Грей бокс (gray box)** - сочетание тестирования "черного ящика" и тестирования "белого ящика".

30. **Девелопер (developer)** - разработчик, программист. Человек, занимающийся программированием, выполняет разработку программного обеспечения для программируемых устройств.

31. **Деклайнд (declined)** - от decline: отказаться.

32. **Деплоймент (deployment)** - развертывание ПО. Развертывание программного обеспечения - это все действия, которые делают программную систему готовой к использованию. Данный процесс является частью жизненного цикла программного обеспечения.

33. **Десижн тейблс (decision tables)** - таблицы решений.

34. **Дескрипшн (description)** - описание.

35. **Десктоп (desktop)** - так называют персональный компьютер. Также называют место работы юзера, ведь по умолчанию desktop имеет путь C:\Users\User\Desktop.

36. **Дефект репорт (defect report)** - то же самое, что и баг-репорт: отчет об ошибке.

37. **Деферд (deferred)** - отложен.

38. **Джира (JIRA)** - система отслеживания ошибок, предназначена для организации общения с пользователями, хотя в некоторых случаях может быть использована для управления проектами.

39. **Домен (domen)** - одним из значений является сфера деятельности заказчика.

40. **Дропдаун (dropdown)** - выпадающий список. Например, для выбора городов из списка.

41. **Эджайл (agile)** - класс методологий разработки программного обеспечения, основанный на итеративной разработке, в которой требования и решения эволюционируют через сотрудничество между самоорганизованными многофункциональными командами. Гибкая разработка - лучшее средство для повышения производительности разработчиков программного обеспечения.

42. **Экспириенс (experience)** - опыт.

43. **Эксплоретари тестинг (exploratory testing)** - исследовательское тестирование, во время которого можно как искать дефекты, так и изучать программное обеспечение.

44. **Эмбедед (embedded)** - встроенный.

45. **Эмэйл (email)** - электронная почта. Популярный сервис в интернете, что делает возможным обмен данными любого содержания (текстовые документы, аудио, видео-файлы, архивы, программы).
46. **Энвайремент (environment)** - среда.
47. **Эриа (area)** - участок, плоскость.
48. **Эрроп гесинг (error guessing)** - метод тестирования, в котором тест-кейсы используются для поиска ошибок в программах на основе опыта предыдущего тестирования. Фактически, это метод угадывания ошибок.
49. **Эс Кью Эль (SQL)** - язык структурированных запросов - декларативный язык программирования для взаимодействия пользователя с базами данных, применяется для формирования запросов, обновления и управления реляционными БД, создания схемы базы данных и ее модификации, системы контроля за доступом в базу данных.
50. **Эсайнт (assigned)** - порученный.
51. **Эттачмент (attachment)** - прикрепление, например, в электронной почте.
52. **Жизненный цикл разработки ПО (software development life cycle)** - совокупность отдельных этапов работ, проводимых в заданном порядке в течение периода времени, который начинается с решения вопроса о разработке программного обеспечения и заканчивается прекращением использования программного обеспечения.
53. **Зааффектить (от affect)** - повлиять.
54. **Задизайнить (от design)** - дословно от дизайн.
55. **Залогиниться (от login)** - войти в систему с определенным набором логина и пароля.
56. **Зарепортировать (от report)** - написать отчет.
57. **Иквиваленс партишн (equivalence partition)** - метод тестирования программного обеспечения, который делит входные данные блока программного обеспечения на разделы эквивалентных данных, на основе которых составляются тесты. В принципе, тест-кейсы разрабатываются для покрытия каждого раздела по крайней мере один раз.
58. **Инспекшенс (inspections)** - инспекция, экспертиза.
59. **Интеграционное тестирование (integration testing)** - это фаза тестирования программного обеспечения, при которой отдельные модули программы комбинируются и тестируются вместе, во взаимодействии. Интеграционное тестирование выполняется после модульного тестирования и перед верификацией и валидацией ПО. Если рассматривать этот процесс как систему, то на вход ей подаются модули, которые уже прошли модульное тестирование; затем модули группируются в большие части, выполняются тесты, предусмотренные планом, а на выходе получается интегрированная система, готова к системному тестированию.

60. **Интродакшн (introduction)** — вступление. Также используется в значении «представление\знакомство с системой».
61. **Информал (informal)** - неформальный.
62. **Кастомер (customer)** - заказчик.
63. **Кэш (cache)** - временное хранилище для файлов, которые часто используются. Обычно им пользуются веб-браузеры для ускорения загрузки страниц.
64. **Кликабельна (clickable)** - есть возможность нажать с помощью указателя мыши.
65. **Кодирование, коддинг (coding)** - процесс написания программного кода.
66. **Коммон сенс (common sense)** - здравый смысл.
67. **Контакт поинт (contact point)** - дословно: точка контакта.
68. **Контрол (control)** - в языках программирования так называют элемент управления: кнопка, текстовое поле, флажок и т.
69. **Копи-паст (copy / paste)** - скопировать, а затем вставить.
70. **Критикал (critical)** - критическое.
71. **Лид (lead)** - руководитель.
72. **Линк (link)** - ссылка на страницу.
73. **Лoad тестинг (load testing)** - это просто форма тестирования производительности. Оно обычно проводится для того, чтобы оценить поведение программы (приложения) с заданной ожидаемой нагрузкой. Этой нагрузкой может быть, например, количество пользователей, которые будут одновременно работать с программой. Такой вид тестирования позволяет получить время отклика всех важнейших бизнес-транзакций.
74. **Лог (log)** - журнал работы программы.
75. **Лук эраунд (look around)** - смотреть вокруг.
76. **Манки джоб (monkey job)** - как правило, простая работа, которая не требует использования сложных подходов и методик. Также употребляется в значении элементарной и рутинной работы.
77. **Мануальный (от manual)** - ручной.
78. **Медиум (medium)** - умеренный, средний.
79. **Мейджор (major)** - более важный, основной.
80. **Минорный (minor)** - второстепенный, не очень важен.
81. **Митинг (meeting)** - встреча.
82. **Мокап (mock up)** - макет.
83. **Натянули ПО** - использовали готовое ПО, без написания для нового аппаратного обеспечения.
84. **Нотификация (notification)** - напоминание. уведомления.

85. **Пассед (passed)** - пройденный, выполненный.
86. **Перфоменс тестинг (performance testing)** - это тестирование, которое проводится с целью определения, как быстро работает программа или ее часть под некоторой нагрузкой. Тестирование производительности пытается подсчитать производительность на стадии.
87. **Постить (от post)** - опубликовать.
88. **Пофиксить (от fix)** - исправить ошибку.
89. **Прекондишн (precondition)** - предпосылка.
90. **Продакт овнер (product owner)** - владелец продукта.
91. **Продакт стайл гайд (product style guide)** - документ, который описывает правильное использование графических и функциональных элементов платформы для разработки программного обеспечения под эту платформу (пример, гайд от компании Apple, которая жестко регламентирует как должны выглядеть и функционировать программы для операционной системы iOS).
92. **Продакшн (production)** - производство. Так обычно называют выпуск готового продукта.
93. **Проектный менеджер (project manager)** - менеджер проекта.
94. **Проинсталлировать (от install)** - установить ПО на персональный компьютер или сервер.
95. **Профайл (profile)** - профиль, набор данных.
96. **Профит (profit)** - прибыль, польза.
97. **Прошивка (firmware)** - ПО аппаратных устройств, например роутеров.
98. **Рандом (от random)** - случайный.
99. **Ревью (review)** - проверка.
100. **Регрессивное тестирования (regression testing)** - общее название для всех видов тестирования программного обеспечения, направленных на выявление ошибок в уже протестированных участках исходного кода. Такие ошибки - когда после внесения изменений в программы перестает работать то, что должно работать, - называют регрессивными ошибками.
101. **Реквайрмент (requirement)** - требование.
102. **Релиз (release)** - релиз, выпуск. Издание продукта, готового к тиражированию. Это стабильная версия программы, прошла все предыдущие стадии разработки, в которых исправлены основные ошибки, но существует вероятность появления новых, ранее не замеченных недостатков.
103. **Реоупен (reopen)** - открытый повторно.
104. **Репозиторий (repository)** - хранилище. Специальный сервер, с которого можно загрузить программное обеспечение. На сервере хранится архив программных продуктов,

которые доступны для загрузки.

105. **Репорт (report)** - отчет.

106. **Репродьюсд (reproduced)** - повторен, воспроизведен.

107. **Роутер (router)** - роутер, еще называют маршрутизатором. Это электронное устройство, используемое для соединения двух или более сетей и руководит процессом маршрутизации.

108. **Сабмиттед (submitted)** - отправлен, оставленный.

109. **Саммари (summary)** - итог, резюме.

110. **Северити (severity)** - строгость, влияние на систему в целом. **ПР - в основном Severity переводится как степень серьезности или степень "тяжести" ошибки.**

Пометка Редактора Тани — Severity может использоваться в значении *важность*, т.к. есть фраза priority and severity (приорити и северити (приоритетность и важность)).

111. **Сервер (server)** - в компьютерной терминологии срок может касаться отдельного компьютера или программы. Главным признаком в обоих случаях является способность машины или программы подавляющее количество времени работать автономно, без вмешательства человека, реагируя на внешние события согласно установленному программному обеспечению. Или же центральный компьютер, который выполняет роль главного в сети.

112. **Сет (set)** - набор.

113. **Системное тестирование (system testing)** - является одним из уровней тестирования программного обеспечения. Системное тестирование тестирует интегрированную систему для проверки соответствия всем требованиям. Проверка полноты и правильности документации пользователя является важной частью системного тестирования. Все тестовые комбинации должны разрабатываться только с использованием документации пользователя.

114. **Скоуп (scope)** - объем, масштаб.

115. **Скрам (scrum)** - методология управления проектами для гибкой разработки программного обеспечения. Скрам четко делает акцент на качественном контроле процесса разработки.

116. **Скрам мастер (scrum master)** - роль в Скрам, человек, который следит за выполнением всех процессов Скрам.

117. **Скрипт (script)** - это программа, которая автоматизирует некоторое задание, которое без сценария пользователь делал бы вручную, используя интерфейс программы.

118. **Скриншот (screen shot)** - копия изображения экрана сохранена в графическом формате: png, jpg, и тому подобное.

119. **Солюшн (solution)** - решение.

120. **Стейт транзишн тейбл (state transition table)** - таблица переходов программы из

одного состояния в другое.

121. **Степ (step)** - один шаг.

122. **Степс ту репродьюс (steps to reproduce)** - шаги для воспроизведения.

123. **Стракче (structure)** - структура.

124. **Стресс тестирование (stress testing)** - Стрессовое тестирование обычно используется для установления границ пропускной способности программы. Этот тип тестирования проводится для определения надежности системы при экстремальных или непропорциональных нагрузках и отвечает на вопрос о достаточную производительность системы в случае, если текущее нагрузки значительно превысит ожидаемый максимум.

125. **Таск (task)** - задача.

126. **Темплейт (template)** - шаблон.

127. **Тест екзекьюшн (test execution)** - выполнение тестов.

128. **Тест кейс (test case)** - это документ, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или ее части. Целенаправленно создана тестовая ситуация.

129. **Тест сьют (test suite)** - набор тест кейсов или отдельных тестов.

130. **Тим (team)** - команда.

131. **Тим лид (team lead)** - лидер команды, обычно наиболее опытный.

132. **Трак (trac)** - свободное браузерное приложение для управления проектами.

133. **Трейсебилити матрикс (traceability matrix)** - матрица отслеживания, таблица зависимостей требований к тест-кейсам, используется для отслеживания покрытия требований тестами.

134. **Тулы, тулзы (tools)** - утилита, дополнительное программное обеспечение.

135. **Фейлд (failed)** - неудачный.

136. **Фидбэк (feedback)** - отзыв, обратная связь. Также может использоваться как «ответ».

137. **Фиксед (fixed)** - исправлен.

138. **Фича (от feature)** - дословно: особенность. В языке ИТ какая-то отдельная функция программы.

139. **Фрилансер (freelancer)** - вольнонаёмный, который сам ищет себе проекты, может одновременно работать на несколько фирм. В Украине фрилансерами преимущественно называют людей, выполняющих любую работу через сеть интернет удаленно.

140. **Фронт энд (front end)** - говоря о front-end и back-end, программисты обычно имеют в виду отделение интерфейсной части пользователя от программной логики. **Примечание:** Back-end - это все что связано с написанием серверных скриптов. Front-end - это интерфейс взаимодействия между пользователем и основной программно-аппаратной частью.

141. **Функциональное тестирование (functional testing)** - выявление несоответствий между реальным поведением реализованных функций и ожидаемым поведением согласно спецификации и требований. Функциональные тесты должны охватывать все реализованные функции с учетом наиболее вероятных типов ошибок. Тестовые сценарии, сочетающие отдельные тесты, ориентированные на проверку качества решения функциональных задач.
142. **Чек лист (checklist)** - перечень, список. Это документ, описывающий что должно быть протестировано. При этом чек-лист может быть абсолютно разного уровня детализации. На сколько детальным будет чек-лист зависит от требований к отчетности, уровня знания продукта сотрудниками и сложности продукта.
143. **Юзер интерфейс, юай (user interface, UI)** - средство удобного взаимодействия пользователя с информационной системой.
144. **Юзер стори (user story)** - это одно или несколько предложений обычным языком предметной области, которые описывают, чего пользователь хочет достичь. Истории пользователя применяются в гибких методологиях для выяснения базовых функций, которые будут реализовываться.
145. **Юзерфрендли (user friendly)** - интерфейс, удобный и понятный пользователю.
146. **Юнит тест (unit test)** - это метод тестированию программного обеспечения, который заключается в отдельном тестировании каждого модуля кода программы. Модулем называют наименьшую часть программы, которая может быть протестированной.
147. **upwork.com** - фриланс биржа. Примечание, odesk. com и elance. com - этих ресурсов больше нет.
148. **ISTQB - International Software Testing Qualification Board.** Международная коллегия тестирования программного обеспечения.
150. **QA (Quality Assurance)** - эксперт по обеспечению качества, что, как правило, задействован в активности, которая предотвращает возникновение ошибок.
151. **QC (Quality Control)** - эксперт по контролю качества, что, как правило, задействован в поиске ошибок и их исправлении.
152. **Software architecture document** - документ, который описывает архитектуру программы и подходы и технологии, которые будут использоваться для ее разработки.
153. **Tester** — тестировщик ПО.
154. **XML** - предложенный консорциумом World Wide Web (W3C) стандарт построения языков разметки иерархически структурированных данных для обмена между различными приложениями, в частности, через Интернет. Есть упрощенным подмножеством языка разметки SGML. XML документ состоит из текстовых знаков, и пригоден к чтению человеком.