TEST PLAN OUTLINE

В стандарте IEEE 829 перечислены пункты, из которых должен состоять тест-план.

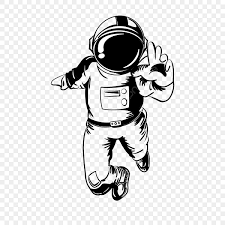
Тест план

1. **Test plan Identifier** - Идентификатор тест плана.

ООО «Котятки». Тест план на ПО «Лежаночка» ver 1.0

*Название компании, логотип компании*

*Logo*

**

1. **References** - Ссылки

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
| 22.06.2023 | 1.0 | Тест план на ПО «Лежаночка» | Иванов И.И. |
| 26.06.2023 | 1.0 | Прикрепление ссылки на ПО «Лежаночка» | Петров А.М. |

1. **Introduction** - Наша компания осуществляет функциональное и нефункциональное тестирование для выявления ошибок в ПО «Лежаночка» до выпуска релиза. Мы выполняем тщательное тестирование заявленных функциональных возможностей, чтобы помочь достичь целей бизнеса для вашего ПО.

*Введение. Пояснительная записка для клиента. Виды тестирования для ПО.*

1. **Test items** - Объекты тестирования
2. Объект тестирования: сайт лежаночка.ru
3. Предполагается провести тестирование на объекты:
4. Форма регистрации
5. Покупка
6. UI
7. Бронирования
8. подписки
9. Позитивное и негативное тестирование по заявленным видам тестирования.
10. **Software Risk Issues** - Проблемы и риски
11. Отпуска
12. Больничные
13. Отсутствие оборудования
14. Профилактика оборудования
15. Ураган котят
16. Отсутствие финансирования со стороны клиента (один из критериев для остановки тестирования)
17. **Features to be Tested** - Функции, которые нужно протестировать
18. Форма регистрации – высокий приоритет
19. Покупка – высокий приоритет
20. UI – средний

Не будет протестировано нагрузочное тестирование и тестирование безопасности ввиду отсутствия необходимых ресурсов.

*Фичи - функции, которые нужно протестировать*

1. **Features not to be Tested** - Функции, которые не нужно тестировать
2. Бронирование
3. Подписки

Исходя из рисков и запланированных отпусков данный функционал не будет протестирован в данном релизе.

1. **Approach** - Подходы
2. Метод черного ящика
3. Исследовательское
4. UI/UX
5. Смоки тест
6. Кроссбраузерность
7. Функциональное
8. **Item Pass/Fail Criteria** - Критерии прохождения тестов

Критерии начала:

1. Законченная документация
2. Готовность тестового стенда
3. Код функции доставлен на тестовую площадку (деплой)
4. Составленные тест – кейсы
5. Предоставление специализированного ПО для тестирования

Критерии окончания:

1. Все тесты проведены
2. Исправлены все баги
3. Отмечены все неисправленные баги (некритичные)
4. Отсутствие финансирования
5. Учтены пожелания заказчика (требования)
6. Все точки отказа задокументированы (например, из-за неисправности железа)
7. **Suspension Criteria Resumption Raquirements** - Критерии остановки и возобновление тестирования

Критерии остановки:

1. Изменение требований
2. Блокирующие баги
3. Просьба (требование) заказчика
4. Отказ оборудования (форс мажор)

Критерии возобновления:

1. Блокеры исправлены
2. Исправность оборудования
3. Просьба заказчика
4. Актуальная версия требований
5. **Test Deliverables** - Результаты тестирования:
6. Количество завершенных тестов 99%
7. Количество найденных багов: 5
8. Написано 100 тест кейсов
9. Количество исправленных багов: 5
10. Количество фича - реквестов: 8

*Рекомендованный список feature - requests для улучшения продукта*

1. **Remaining Test Tasks** - Оставшиеся задачи по тестированию:
2. Безопасности – при наличии свободных sp
3. Нагрузочное – при наличии JMeter

*(см. пункт 6)*

***Repoint – условная единица оценки задачи***

1. **Environmental Needs** - Требования среды и тестового окружения

|  |  |
| --- | --- |
| Браузеры | Google, FF, Safari |
| OS | Win 10 - 11, MacOs, Unix |
| Мобильные платформы | Android, IOS |
| Разрешение экрана desktop | 1280х768, 1929х1080 |
| Мобильные дисплеи | 480х800, 640х960 |
| ПО | BrowserStack |

1. Тестовые стенды
2. Наличие эмуляторов среды
3. **Staffing End Training Needs** - Требования по части кадров и их обучения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Количество | Уровень | Задачи | Время | Оплата | Skill Up |
| 1 | Senior | Управление командой, планирование | Full Time | 1000$/ч | Посещение семинаров |
| 2 | Middle | AQA, наставничество | Full Time | 700$/ч | Консультация с Senior (one To one) |
| 3 | Junior |  |  |  |  |

1. **Responsibilities** - Распределение обязанностей

|  |  |  |
| --- | --- | --- |
| Петров ИИ | Сеньоры | Управление командой, планирование |
| Иванов НИ | Мидл | AQA |
| ФИО | Джуниор |  |
| ФИО | Джун |  |
| ФИО | Джун |  |

1. **Schedule** - Расписание

|  |  |  |  |
| --- | --- | --- | --- |
| Документация | 22.06-26.06 | Петров | 8ч |
| Автотесты | 22.06-23.09 | Иванов | 200ч |
|  |  |  |  |

*Составление документации*

1. **Plaining Risks and Contingencies**
2. **Approvals**
3. **Glossary**

|  |  |
| --- | --- |
| **Бэклог** | это перечень рабочих задач, которые необходимо выполнить команде. Бэклог — это термин из Agile и Scrum (методологии разработки). Составляется он на основе требований заказчика продукта. Наиболее важные задачи расположены в самом начале бэклога, для того чтобы команда понимала, какую работу нужно выполнить в первую очередь. |
| **Валидация** | это процесс проверки того, соответствует ли разрабатываемое программное обеспечение заданным требованиям и спецификациям. |
| **Итерация (Iteration)** | это один повтор выполняемой в цикле операции. |
| **Майнд-карта (Mind Map, интелектуальная карта)** | Карта представляет собой диаграмму с центральным значением (например, название проекта) и исходящими от него линиями, на которых могут быть расположены символы, изображения и другие обозначения. Mind Maps не только отображают список задач, но и показывают взаимосвязь элементов |
| **Метрика** | качественный или количественный показатель, который отражает ту или иную характеристику и уровень успешности продукта |
| **Спринт** | промежуток времени, достаточный для выполнения запланированной совокупности операций Scrum, целью которой является создание инкремента бизнес-продукта. Жёстко фиксирован по времени. Длительность одного спринта от 1 до 4 недель. Чем короче спринт, тем более гибким является процесс разработки, релизы выходят чаще, быстрее поступают отзывы от потребителя, меньше времени тратится на работу в неправильном направлении, но много времени тратится на митинги планирования спринта, ретроспективы. С другой стороны, при более длительных спринтах команда (Scrum Team) уменьшает издержки на совещания, демонстрации продукта и т. п. Разные команды подбирают длину спринта согласно специфике своей работы, кросс-функциональности команд и требований, часто методом проб и ошибок. Для оценки объёма работ в спринте можно использовать предварительную оценку, измеряемую в баллах истории. Предварительная оценка длины спринта фиксируется в бэклоге проекта (product backlog; см. далее). |
| **Service Level Agreement, SLA (Соглаше́ние об у́ровне предоставле́ния услу́ги)** | термин методологии ITIL, обозначающий формальный договор между заказчиком (в рекомендациях ITIL заказчик и потребитель — разные понятия) услуги и её поставщиком, содержащий описание услуги, права и обязанности сторон и, самое главное, согласованный уровень качества предоставления данной услуги. |
| **Тикет-система** | это система, которая обеспечивает взаимодействие ИТ-подразделения и пользователей. «Ticket» переводится как «билет», что означает, что все запросы подаются в готовой форме. Одним из главных достоинств тикет-системы является систематизация запросов. |
| **API (Application Programming Interface)** | что значит программный интерфейс приложения. В контексте API слово «приложение» относится к любому ПО с определенной функцией. Интерфейс можно рассматривать как сервисный контракт между двумя приложениями; |
| **ad-hoc testing** | вид тестирования, который выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев |
| **Assign (ассайнить)** | поручить какое-либо задание кому-то; |
| **Attach (аттачить)** | приложить документ или фотографию к тексту; |
| **Automated (автоматизированное)** | тестирование с помощью специально написанных автоматических тестов; |
| **Back-end (бэкэнд)** | это внутренняя часть продукта, которая находится на сервере и скрыта от пользователей. Для её разработки могут использоваться самые разные языки, например, Python, PHP, Go, JavaScript, Java, С#; |
| **Backup (бэкап, бэкапить)** | замещать или создавать дополнительную копию (в большинстве случаев речь идет о базах данных); |
| **Bug (ошибка)** | дефект ПО, из-за которого ожидаемый результат работы приложения не сходится с фактическим; |
| **Bug report (баг-репорт)** | документ, содержащий информацию о выявленном дефекте; |
| **Build (билд)** | конечный вид ПО или конкретного компонента, который можно тестировать, зачастую его еще называют сборкой многих задач программистов в одну общую; |
| **CI/CD (Continuous Integration, Continuous Delivery** | непрерывная интеграция и доставка) — это технология автоматизации тестирования и доставки новых модулей разрабатываемого проекта заинтересованным сторонам (разработчикам, аналитикам, инженерам качества, конечным пользователям и др.) |
| **debug** | Программа-отладчик, которую используют для проверки и отладки выполняемых файлов. Использовалась при операционной системе DOS. Под более поздние версии операционных систем работает через эмулятор DOS и имеет ограниченные возможности. Также иногда называют процесс отладки программы. |
| **entry criteria** | начало тестирования |
| **exit criteria** | прекращение тестирования |
| **feedback(фидбэк)** | обратная связь, отзыв на что-либо, конструктивная и развернутая оценка |
| **Fix (фиксить)** | чинить ПО, чинить баги; |
| Front-end (фронтэнд) | это клиентская часть продукта (интерфейс, с которым взаимодействует пользователь). В случае с сайтом, её формирует и выводит на экран браузер, который работает с хтмл, CSS и JavaScript; |
| **Kanban** | система постановки задач и организации рабочих процессов для эффективного достижения поставленных целей |
| **Local (локаль)** | адаптация ПО под регион; |
| **Manual (ручное)** | тестирование вручную; |
| **Postman** | это платформа API, позволяющая разработчикам проектировать, создавать, тестировать и повторять свои API |
| **Pre-production** | среда, максимально приближенная к реальной среде использования продукта |
| **Production** | реальная среда использования продукта |
| **Quality Assurance (QA, обеспечение качества)** | работа по предотвращению появления багов; |
| **Quality Control (QC, контроль качества)** | работа по оценке соответствию ПО требованиям; |
| **Release (релиз)** | финальный выпуск программы; |
| **REST API** | это способ взаимодействия сайтов и веб-приложений с сервером. Его также называют RESTful. |
| **resumption criteria** | возобнавление тестирования |
| **Scrum** | набор правил для организации гибкого рабочего процесса, который заключается в командном подходе, работе итерациями, фокусировке на цели каждой итерации и нестандартном распределении обязанностей внутри коллектива |
| **SoapUI** | приложение с открытым исходным кодом для тестирования веб-сервисов сервис-ориентированных архитектур и передачи состояний представлений. |
| **Specification (спек, спека, спецификация)** | техническое описание ПО, на которое ориентируется тестировщик ПО; |
| **Staging** | среда, в которой происходит проверка перед боем. Её особенностью является максимальное приближение к условиям боевой среды, что дает возможность полнее протестировать то, что происходит. Обычно это то место, куда идут менеджеры, тестировщики, заказчики. Часто стейджинг выполняет сразу две задачи — проверку конкретных фич от разработчиков и окончательный прогон приложения перед релизом. |
| **suspension criteria** | приостановка тестирования |
| **Swagger** | это набор инструментов, который позволяет автоматически описывать API на основе его кода. API — интерфейс для связи между разными программными продуктами, и у каждого проекта он свой. Документация, автоматически созданная через Swagger, облегчает понимание API для компьютеров и людей; |
| **Test Artefact (тестовый артефакт)** | документ, который используется при проведении тестирования; |
| **Test Case (тест-кейс)** | пошаговый план проверки ПО; |
| **Test Coverage (тестовое покрытие)** | условная площадь ПО, которая покрывается всеми тестами; |
| **Test strategy** | стратегия тестирования |
| **Test Suite (тестовый набор)** | несколько объединенных тест-кейсов для проверки одного типа; |
| **Tool (тул)** | вспомогательная программа |
| **Validation (валидация)** | оценка работоспособности ПО; |
| **Verification (верификация)** | сравнение заданных заказчиком требований и ПО. |
| **Requirement (требование)** | описание того какие функции и с соблюдением каких условий должно выполнять приложение в процессе решения полезной для пользователя задачи |
| **What, not how** | что, а не как |
| **Метод trim** | удаляет пробельные символы с начала и конца строки. Пробельными символами в этом контексте считаются все собственно пробельные символы (пробел, табуляция, неразрывный пробел и прочие) и все символы конца строки (LF, CR и прочие) |