Классификация видов тестирования:

|  |  |  |
| --- | --- | --- |
| **1. Функциональное functional** | | |
| 1.1. Функциональное functional | сравнительный анализ спецификации и функциональности компонента / системы |  |
| 1.2. Безопасности security | оценка уязвимости к атакам |  |
| 1.3. Взаимодействия interoperate | Оценка качества взаимодействия компонент программы/всего приложения с другими приложениями/ПО |  |
| **2. Не функциональное** | | |
| 2.1. Доступности Accessibility | степень легкости, с которой пользователи с ограниченными способностями могут использовать систему |  |
| 2.2. Требований Requirements | проверка требований на соответствие основным атрибутам качества |  |
| 2.3. Конфигурационное configuration | проверка работы ПО при различных конфигурациях системы. |  |
| 2.4. На отказ и восстановление failover & recovery | тестирование при помощи эмуляции отказов системы или реально вызываемых отказов в управляемом окружении |  |
| 2.5. Пользовательского интерфейса GUI | взаимодействие с системой через графический интерфейс пользователя: правописание информации; расположение элементов; соответствие названий форм их назначению; унификация стиля, цвета, шрифта; окна сообщений; размеры окна, поведение курсора, горячие клавиши |  |
| 2.6. Удобства использования UI | степень понятности, легкости в изучении и использовании, привлекательности в эксплуатации (визуальное оформление, навигация, логичность, наличие обратной связи и др.), т е практичность |  |
| 2.7. Установки installation | тестирования установки программного продукта. (проверка интерфейса, навигации); тестирование механизма лицензирования и функций защиты от пиратства; стабильность приложения после установки. |  |
| 2.8. Производительности performance | скорости реакции программного продукта на внешние воздействия при нагрузке. |  |
| 2.9. Локализации Localization | адаптация продукта к языковым и культурным особенностям конкретного региона, отличного от того, в котором разрабатывался продукт |  |
| 2.10. Совместимости Compatibility | работоспособность приложения в различных средах (браузеры и их версии, ОС: их типы, версии и разрядность). Виды тестирования совместимости: 1) кроссбраузерное (браузеры или их версии), 2) кроссплатформенное (ОС или их версии). |  |
| **3. связанное с изменениями** | | |
| 3.1. Сборки build | соответствие выпущенной версии критериям качества для начала тестирования. |  |
| 3.2. Регрессионное regression | проверка ПП на отсутствие ошибок уже работающего кода при вводе новых ф-цион-тей или модификаций. Т е проверить что раньше работало – продолжает работать. |  |
| 3.3. Санитарное sanity | узконаправленное для определения работоспособности определенной части системы. |  |
| 3.4. Дымовое smoke | 1. приложение стартовало, 2. ссылки рабочие (не битые), 3. регистрация работает, 4. отчет загружается, 5. файлы грузятся. По 1у заходу.  поверхностное тестирование для определения пригодности сборки для дальнейшего тестирования, должно покрывать базовые функции ПО |  |
| 3.5. Повторное | выполнение тест-сценариев, выявившие ошибки во время последнего запуска, для подтверждения успешности исправления этих ошибок. Повторное – частный случай Регрессионного тестирования. |  |
| **4. Производительности** | | |
| 4.1. Производительности Perfomans | скорости реакции ПП при нагрузке. |  |
| 4.2. Нагрузки Load | сохранять качество при нагрузке в допустимых пределах и их превышении (количестве параллельных пользователей и/или операций). |  |
| 4.3. Стресс Stress | поведение приложения при нештатных изменениях нагрузки или в состоянии ограниченных ресурсов: память или доступ к серверу |  |
| 4.4. Стабильности и надежности Stability / Reliability | способность выполнять свои функции при длительном тестировании со средним уровнем нагрузки |  |
| 4.5. Объемное Volume | оценку производительности при увеличении объемов данных в БД приложения |  |
| 4.6. отказоустойчивость | проверка способности ПП задействовать механизмы по предотвращению нарушения работы, производительности и повреждения данных в критических ситуациях |  |
| 4.7. Масштабируемость | увеличивать показатели производит-сти при росте ресурсов |  |
| 4.8. конкурентное | обрабатывать большое количество одновременных запросов, что вызывает конкуренцию между ними (запросами) за ресурсы (БД, память, канал передачи данных) |  |

Дополнительная классификация видов тестирования:

|  |  |  |
| --- | --- | --- |
| **1. по состоянию** | | |
| 1.1. статическое Static | документации |  |
| 1.2. динамическое Dynamic | процесс установки, регистрация, загрузка файлов, дебаг. |  |
| **2. по знанию с-мы** | | |
| 2.1. Black-box | без знания внутренней структуры и компонентов системы (нет доступа к внутренней структуре и коду приложения) |  |
| 2.2. White-box | анализ внутренней структуры компонентов/ системы (есть доступ к внутренней структуре и коду приложения) |  |
| 2.3. Gray-box | Расширение Black-box.  комбинация методов белого и черного ящика, состоящая в том, что к части кода архитектуры есть доступ, а к части кода – нет |  |
| **3. по степени автоматизации** | | |
| 3.1. Ручное Manual | тест-кейсы выполняются вручную без средств автоматизации |  |
| 3.2. Автоматизированное Automated | набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования |  |
| 3.3. Semi | Тест-кейсы частично выполняются вручную, частично автоматизированы |  |
| **4. по степ подготовленности** | | |
| 4.1. по документам Formal | по подготовленным тестовым сценариям, руководству по осуществлению тестов |  |
| 4.2. интуитивное  Ad hoc | без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев. |  |
| 4.3. Исследовательское | метод проектирования тестовых сценариев во время выполнения этих сценариев. |  |
| **5. по время проведения** | | |
| 5.1. Alpha | имитация реальной работы с системой штатными разработчиками на ранней стадии разработки, но не проектной командой |  |
| 5.2. Beta | клиентами на внешней стороне с целью определения удовлетворения компонента или системы требованиям клиента и вписывается ли он в бизнес-процессы |  |
| 5.3. Приемочное User Acceptance | определения соответствия системы критериям приёмки и дать возможность пользователям определить можно ли принимать систему |  |
| **6. Признак позитивности (критерию) сценариев** | | |
| 6.1. Positive | Реакция на ввод валидных значений |  |
| 6.2. Negative | Реакция на ввод не валидных значений |  |
| **7. степени изолированности** | | |
| 7.1. Модульное Component/unit | тестируются отдельные части/модули системы |  |
| 7.2. Интеграционное Integration | тестируется взаимодействие между отдельными модулями |  |
| 7.3. Системное System | тестируется работоспособность системы в целом |  |
| 7.4. ПолноеAcceptance | Тесты покрывают всесценарии тестирования: 1. Ввод корректных/некорректный значений, 2. Исключительные ситуации, 3. Комбинация входных параметров, 4. Классов эквив-ти и граничных значений |  |