

Test Strategy:

1. Introduction
2. Scope
3. Test Environment
4. Минимальные требования
5. Test Exclusions
6. Quality Metrics
7. Test Types
8. Test Tools
9. Test Phases
10. Test Phase Acceptance
11. Defect Reporting
12. Test Summary Reporting

Introduction

Заказчик занимается строительством и обслуживанием торгово-развлекательных центров по всему миру. Постоянно возрастающее количество объектов требует внедрения эффективных информационных технологий в различные бизнес-процессы Заказчика, в частности в процессы учёта и обслуживания телекоммуникационного оборудования Заказчика.

Тестирование будет охватывать функциональные возможности для этого продукта подробно описанные в документе SMB Solution Design со спецификациями требований.

Установка будет протестирована на различных браузерах, как описано в Спецификации требований. Тестирование набор критических функций, чтобы определить, будет ли код работать во всех браузерах.

Scope

Это стратегия тестирования для Shopping-Mall Business Corp. (далее – Заказчик). Этот документ должен быть заполнен и использоваться командой тестирования проекта как руководством по тестированию этого проекта. Менеджер по обеспечению качества должен рассмотреть и утвердить окончательную версию документа Стратегии тестирования.

Test Environment

Тестирование подтверждает, что система будет функционировать на всех требуемых браузерах:

Google Chrome (или Iron Portable),

Firefox,

IE версии выше 7.

Минимальные требования

Аппаратное обеспечение:

- Компьютер с процессором как минимум Intel® i3 (процессор 1600 мегагерц для Windows® 10)
- Серверный компьютер с процессором Pentium 4 / 2400/2гб ОЗУ для Windows Server 2003®
- 300 МБ доступного дискового пространства
- Windows-совместимая мышь
- Графическая карта VGA или SVGA, совместимая с Windows

Test Exclusions

Не требуются Мобильное, Регрессионное и тестирование Производительности.

Quality Metrics

Requirement Coverage rate % = (Количество требований покрытых тестами / Количество требований) * 100.

Execution rate % = (Количество пройденных тестов / Количество запланированных тестов) * 100.

Pass rate % = (Количество успешных тестов / Количество пройденных тестов) * 100

Fail rate % = (Количество проваленных тестов / Количество пройденных тестов) * 100

В случае если количество багов с высоким приоритетом и выше не превышает 1% Fail rate и не больше 3шт.

Test Types

Во время тестирования будут применены следующие виды тестирования:

Smoke testing для проверки самых важных функциональностей продукта (Регистрация, Логин, Создание и изменение объектов инвентаря)

Security будет применен для проверки полей Username, Password, Role и воздействие их вариантов на доступ к системе.

UI будет применен для проверки корректного отображения веб-приложения согласно мокапу, проверены сообщения об ошибках и их размещение, а также проверена орфография и ошибки на страницах веб-приложения.

Compatibility (Cross-browser testing) будет использоваться для проверки корректной работы, согласно предъявленным требованиям, на браузерах из требований.

Sanity будет применен для проверки функциональностей продукта не проверенной в рамках Smoke тестирования.

Regression будет использоваться для проверки функциональных и нефункциональных требований, которые могли быть задеты изменениями в программе или окружающей среде, при помощи тест-кейсов использовавшихся для проверки предыдущей версии.

New Feature будет применен для проверки новой функциональности, появившейся в новой версии (build) программы.

Re-test будет использоваться для проверки факта исправления ранее найденных багов.

UAT (Приемочное пользовательское тестирование) будет использоваться как заключительная стадия перед внедрением программного комплекса с привлечением конечных пользователей системы и с применением специально подготовленных сценариев UAT-тестирования.

Grey Box тестирование будет применено для проверки корректной работы базы данных а также для помощи проверки результатов тестов

White Box будет проводиться программистами для создания и проведения Unit тестов.

Positive будет применено для проверки корректной работы программы при использовании валидных данных и сценариев.

Negative будет использоваться для проверки валидности и ограничений, с применением техник, таких как Boundary Value Analysis и Equivalence Class Partitioning.

Test Tools

Для тестирования будут использованы Компьютер с процессором Intel® i3 (процессор 1600 мегагерц для Windows® 10) с установленными браузерами: Google Chrome 44.1, Firefox 55.7, IE 8

- Серверный компьютер с процессором Pentium 4 / 2400/2гб ОЗУ для Windows Server 2003® с развёрнутым веб-ориентированным приложением для доступа по сети внутри организации производящей тестирование.

Test Phases

Planning:

На фазе планирования будет разработана тестовая стратегия.

Входными критериями для входа в эту фазу являются созданные документы SOW и проектная документация. По итогам фазы будет создана тестовая стратегия, тест план.

Preparation:

Этап написания тест кейсов для покрытия функциональных и нефункциональных требований относящихся к продукту и заполнение матрицы покрытия требований.

Критериями входа из фазу подготовки являются составленные и подписанные тестовая стратегия, тест план и техническая документация.

Критериями выхода являются составленные тест кейсы покрывающие требования для проверки заявленного функционала продукта и заполненная матрица покрытия требований.

Test execution:

Этап выполнения тест кейсов тестами.

Входные критерии начала этой фазы являются написанные и утвержденные тест кейсы покрывающее требования.

Для выхода из фазы должны отсутствовать открытые баги с высоким Priority\Severity "critical" или "blocker".

Test Phase Acceptance

Критерии качества считаются исходя из метрик. Качество продукта считается допустимым при прохождении 95% тест-кейсов уровня дымового тестирования и 95% тест-кейсов уровня критического пути при условии устранения 99% дефектов высокого Priority\Severity.

Открытые дефекты с высоким приоритетом должны быть добавлены в раздел "Известные проблемы" в документе "Release Notes". Для каждого из дефектов должен быть предоставлен обходной путь (workaround), если это возможно.

Итоговое покрытие требований тест-кейсами должно составлять 100%.

переход к тесту критического пути допустим только при успешном прохождении 100% тест-кейсов дымового тестирования.

Критерии приостановки тестирования: Тестирование может быть приостановлено в случае, если при выполнении не менее 25% запланированных тест-кейсов более 50% из них завершились обнаружением дефекта со средним или высоким Priority\Severity.

Критерии возобновления тестирования: исправление всех блокирующих дефектов и более 75% дефектов с высоким Priority\Severity обнаруженных на предыдущей итерации.

Критерии завершения тестирования: выполнение 100% запланированных на итерацию тест-кейсов.

Defect Reporting

Тест-кейсы могут принимать следующие статусы:

ToDo (создан) - присваивается тест-кейсу при создании, но он ещё не заполнен. Также присваивается в случае обнаружения ошибки в тест-кейсе или изменении требований влияющих на его выполнение.

Done - полностью заполнен, и готов быть добавленным в регрессионные/smoke cycles.

Not executed (открыт\запланирован) - тест-кейс готов к выполнению и ещё не прогонялся.

In progress (выполняется) - тест-кейс находится в режиме выполнения и ещё не завершён, если выполнение тест-кейса занимает продолжительное время.

Skipped (пропущен) - присваивается отменённому тест-кейсу в случае изменения логики тестирования или нехватки времени

Failed (провален) - присваивается, если во время выполнения тест-кейса обнаружен баг, заключающийся в несоответствии ожидаемого результата выполнения любого из шагов тест-кейса с фактическим результатом этого шага.

Blocked (заблокирован) - присваивается, при невозможности выполнения из-за бага его блокирующего или из-за нехватки времени.

Тест-кейсы могут иметь следующие приоритеты:

critical - критически важные элементы, без которых работа невозможна (например логин, регистрация, инвентарная часть программы)

high - важные элементы, помогающие работать с программой (например удаление объектов, редактирование объектов)

medium - элементы средней необходимости (например поиск, размещение элементов на странице не мешающее работе с программой)

Классификация серьезности (Severity) багов (влияние на бизнес заказчика):

Critical. Наивысший приоритет. Назначается экстренным ситуациям, которые очень негативно влияют на продукт или даже бизнес компании, и тем, что требуют немедленного устранения.

High. Высокий приоритет. Предназначен для багов, которые должны быть устранены в первую очередь.

Low. Обычный приоритет, который определяется по умолчанию. Назначается Багам, которые следует устранить во вторую очередь, в рабочем порядке.

Классификация приоритетов (Priority) багов:

Blocker. Блокирующая ошибка. С появлением, большая часть последующей работы с программой невозможна. Блокирует работу или тестирование других компонентов. Для дальнейшей работы необходимо устранить ошибку.

Critical. Критическая ошибка. Постоянно нарушает работу основной функциональности компонента.

Medium. Значительный дефект. Он усложняет работу основного функционала или делает невозможным использование дополнительных функций.

Low. Незначительный дефект. Этот баг влияет на функционал системы в отношении малой степени или имеет очевидные обходные пути, затрудняет использование дополнительных функций.

Возможные статусы багов:

Open - баг найден и занесён в систему.

Assigned (назначен) (обычно назначается автоматически) - багу назначен разработчик для исправления.

In progress - разработчик фиксирует баг

Code review - разработчик закончил фиксировать баг и залил на проверку другим разработкам (потом Resolved)

или же:

Pending (Deferred) - баг заблокирован другим либо ожидается доп инфо.

Отклонен (Rejected) - баг отклонён из-за неактуальности в связи с изменениями требований или не считается багом.

Дубликат (Duplicate) - баг с такой ошибкой уже был ранее добавлен в систему.

после исправления бага разработчиком:

Resolved (Fixed) - устанавливается разработчиком после исправления бага.

после чего исправление бага тестируется и меняется статус на один из следующих:

Ready for test - build с фиксом заливается на энвайрмент (разработчик пофиксил и залил билд на какой-то энв)

In test - выполнение проверки бага

(Вряд-ли) **Проверен** (Verified) - устанавливается после успешной проверки исправленного состояния бага.

Повторно открыт (Reopened) - баг, который должен был быть исправленным, снова найден.

Закрит (Closed) - баг окончательно устранен и более не требует внимания.

Необходимые для заполнения поля при формировании баг-репорта:

Summary - краткое описание бага, состоящее из названия компонента ПО, краткого описания сути несоответствия между AR и ER, а также краткого описания условий наступления описанного несоответствия

Description - детальное описание бага с шагами из тест-кейса, ER и AR.

Build & platform - версия ПО в которой был обнаружен баг.

URL to reproduce, browser - URL адрес и название браузера с версией.

Priority - срочность из списка.

Severity - влияние на бизнес из списка.

Status - из списка возможных статусов.

Affected component - компонент приложения, в котором был обнаружен баг.

Test Summary Reporting

Всего запланированных тест кейсов: 80

Пройдено тест кейсов: 75

Успешных тест кейсов: 74

Багов найдено: 1

Найден 1 баг со приоритетом "normal" в модуле поиска.

Исходя из полученных данных, Fail rate = 1% с учётом отсутствия "critical" багов продукт можно запускать.