Глава 9
Адаптационное тестирование

- Локализация — это процесс адаптации программного обеспечения для нового места эксплуатации. Изменение языка не является его единственной составляющей. Не менее важна, например, и культурная адаптация.

Вопросы, на которые следует обратить внимание, приступая к тестированию:

- Изменен ли исходный код?

Возможно при адаптации программы достаточно изменить несколько внешних файлов. Однако как правило программа требует более серьезных изменений, касающихся не только перевода надписей и сообщений.

- Встроен ли текст в программный код?

Если продукт должен переводиться на несколько языков, подавляющая часть его текста отделена от программного кода. Но краткие сообщения и критические сообщения об ошибках невозможно или нерационально выносить во внешние файлы.

Увеличение длины текста при переводе с английского языка.

Длина англ. текста	Увеличение при переводе
До 10 символов	101-200%
11-20 символов	81-100%
21-30 символов	61-100%
31-50 символов	41-60%
51-70 символов	31-40%
Более 70 символов	30%

Клавиатура.

- Разные раскладки клавиатуры.
- Клавиша <Alt-Gr>.
- Мертвые клавиши *(dead key).*
- Доступ к символам, которые отсутствуют на клавиатуре.

Загрузка, сохранение, импорт и экспорт символов основного и расширенного набора ASCII

- Создайте текстовые файлы со всеми 255 символами расширенного набора ASCII . Сохраните их.
- Сохраните полный набор символов во всех поддерживаемых программой файловых форматах, попробуйте прочитать эти файлы.
- Проверьте, как выполняется импорт и экспорт проблемных символов.

Работа с текстом

- Сборные сообщения

Отображаемое программой сообщение может собираться из нескольких фрагментов. Однако если перевести фрагменты сообщения, а потом их объединить, результат может оказаться неприемлемым.

- Правила переноса.
- Правописание

Если в программный продукт включена функция проверки правописания, убедитесь, что она работает правильно.

- Порядок сортировки
- Преобразование теста к верхнему и нижнему регистру
- Правила подчеркивания

Форматы данных и опции настройки

- Формат времени.
- Десятинные и порядковые разделители в числах.
- Символы # и №.
- Отрицательные числа.
- Денежные символы.
- Единицы измерения.

Специфика культур

- Изображения, связанные с конкретной культурой
- Выходные данные, связанные с конкретной культурой
- Совместимость с местными продуктами

Автоматизированное тестирование

- Разработчик тестов должен знать о тестировании и вопросах локализации гораздо больше, чем любой, кто тестирует программу вручную.
- Тестовые сценарии сами могут быть источниками ошибок. Поэтому их следует тщательно отлаживать, прежде чем применять для тестирования локализованной версии программного продукта

Глава 10

Тестирование документации

- Преимущества хорошей документации.
- Цели тестировщика документации.
- Как тестирование документации повышает надежность программного продукта.
- Распределение персонала.
- Руководство пользователя: стадии разработки.
- Тестирование интерактивной справки.

Преимущества хорошей документации

- Легкость использования.
- Снижение стоимости технической поддержки.
- Повышение надежности.
- Облегчение сопровождения.
- Упрощение установки.
- Коммерческий успех.
- Достоверность информации.

Цели тестировщика документации

- Уделить внимание точности, полноте, ясности, простоте использования документации.
- Произвести тщательную проверку соответствия документации реальной структуре и поведению программы.
- Проверить, не пропущены ли в документации какие-нибудь функции продукта.

Как тестирование документации повышает надежность программного продукта

- Вы найдете гораздо больше ошибок, чем предполагаете.
- Документация является прекрасным источником реальных тестовых примеров.
- Отчетам об ошибках, выявленных в ходе тестирования документации,уделяется особое внимание.

При тестировании документации, необходимо

- В точности выполнить все действия, описанные в руководстве.
- Следовать каждому предложению руководства.
- Проверять каждое утверждение и каждое его очевидное следствие.

Стадии разработки руководства

- *Разработка концепции и базовой структуры.*
- *Подготовка.*
- *Производство.*
- *Публикация.*

Тестирование первой версии руководства

- Проанализируйте структуру документа и как можно раньше составьте свои комментарии.
- Выполните общий анализ руководства.
- Подумайте, какие еще вопросы требуют отдельного освещения.
- Проанализируйте руководство на соответствие концепции продукта.
- Поищите неточности.
- Проверьте сообщения об ошибках.
- Поищите фрагменты руководства, отражающие неудачную конструкцию программы.

Типы документаций

- *Задачно-ориентированная* - перечисляется набор задач, которые можно выполнить с помощью программного продукта, и рассказывается, как это сделать.

- *Функционально-ориентированная* - функции программы описываются сами по себе, нередко просто в алфавитном порядке.

Интерактивная справка

- ***Точность.***

Точность и достоверность интерактивной справки необходимо проверить так же тщательно, как и точность руководства.

- ***Гипертекстовые связи.***

Если в интерактивной справке имеются гиперссылки, все их необходимо проверить.

- ***Указатель.***

Важно проверить правильность и содержание указателя.

- ***Стиль.***

От справки требуется большая четкость и простота изложения. Хорошая справка должна быть задачно-ориентированной.

Глава II
Инструментальные средства тестировщика

- Базовые инструменты тестировщика.
- Автоматизация приемочного и регрессионного тестирования.
- Технологии и инструменты тестирования соответствия стандартам.
- Средства для тестирования "стеклянного ящика".

Базовые инструменты тестировщика

- *Персональный компьютер, терминал или рабочая станция.*
- *Хороший текстовый процессор.*
- *Процессор планов.*
- *Электронная таблица.*
- *Утилиты сравнения файлов.*
- *Просмотровики файлов.*
- *Конвертеры файлов.*
- *Утилиты для создания копий экрана.*
- *Утилиты для поиска текста.*
- *Устройства видеозаписи (VCR).*
- *Диагностические программы.*
- *Таймер.*
- *Система отслеживания проблем.*
- *Программист.*

Откуда берутся регрессионные тесты

- Тесты для граничных условий и другие заранее запланированные тесты.
- Тесты, уже однажды выявившие ошибку.
- Ошибки, выявленные пользователями.
- Наборы тестовых данных, сгенерированных методами случайного подбора.

Способы попадания тестовых данных в программу

- Файлы данных.
- Пакетные файлы.
- Перенаправление ввода.
- Ввод через последовательный порт.
- Перехват и воспроизведение клавиатурного ввода.

Запись вывода тестируемой программы

- Файлы выходных данных.
- Перенаправление вывода в файл.
- Вывод через последовательный порт.
- Перехват экрана.
- Перехват вывода программы с помощью специальных тестированных утилит.

Оценка выходной информации

- Если существует эталонная программа, выполняющая то же самое, можно сравнить их вывод.
- Написать параллельную программу.
- Сформировать библиотеку правильных выходных данных.
- Перехват вывода программы.

Вопросы, которые следует обдумать, планируя автоматизацию тестирования

- **Временные затраты.**

Временные затраты на автоматизацию окупаются только после десятого-одиннадцатого прогона теста.

- **Задержки в тестировании.**

Программистам необходима быстрая ответная реакция тестировщиков.

- **Инерция.**

Если будет изменен интерфейс или формат данных, то огромную часть тестов станет невозможно использовать.

- **Риск пропущенных ошибок.**

За выполнением автоматизированных тестов тестировщик наблюдает не так тщательно, как при выполнении тестов вручную.

- **Частичная автоматизация.**

Следует автоматизировать наиболее подходящие для этого тесты, а также те, которые будут проводиться десятки раз.

Стандарты

- Проблемы с переносимостью.
- Рекурсия.
- Степень вложенности.
- Константы в коде.
- Размер модуля.
- Комментарии.
- Соглашения об именах.
- Форматирование.
- Запрещенные конструкции.
- Запрещенные действия.
- Псевдонимы.
- Преобразование типов данных.

Ошибки, выявляем0ые программой проверки соответствия стандартам

- Неверный синтаксис.
- Смешанные вычисления.
- Переменные, определенные в программном коде, но никогда не используемые.
- Переменные, которые используются до их инициализации.
- Чтение файла до его открытия или после закрытия.
- Код, который никогда не выполняется.
- Бесконечные циклы.

Тестирование "стеклянного ящика"

- **Отслеживание путей выполнения программы**

Существуют специальные программные ***средства мониторинга охвата (coverage monitors),*** *в*ставляющие в код отладочные команды. Эти команды называются ***зондами (probes).***

Недостатки средств мониторинга охвата :

- Средства мониторинга охвата вставляют в программу отладочный код.
- Со средствами мониторинга программа медленнее работает.
- Нашпигованная зондами программа сильно увеличивается в размере.
- Некоторые из таких программных средств сами полны ошибок.

Использование памяти работающей программой

- Часть памяти занимает сам программный код.
- Часть памяти занимают данные.
- Определенная память связана с аппаратным обеспечением.
- Недоступная память.