FUJITSU

shaping tomorrow with you

Тестирование Windows-приложений

November 2014

FUJITSU RUSSIA Global Delivery Centre

FUJITSU

Классификация windows-приложений

- Stand-alone приложения
- Клиент-серверные (client/server) приложения
- Распределенные (fully distributed) приложения

Stand-alone приложения

Определение: к классу stand-alone приложений относятся такие приложения, которые для своей работы не требуют дополнительных компонентов (серверов) и устанавливаются только на 1 компьютере.

Особенности: отсутствует хранилище данных (СУБД), т.е. всю необходимую информацию приложение получает либо от пользователя, либо из файлов данных (локальная файловая БД).

Примеры stand-alone приложений: калькулятор Windows, Word, Excel.

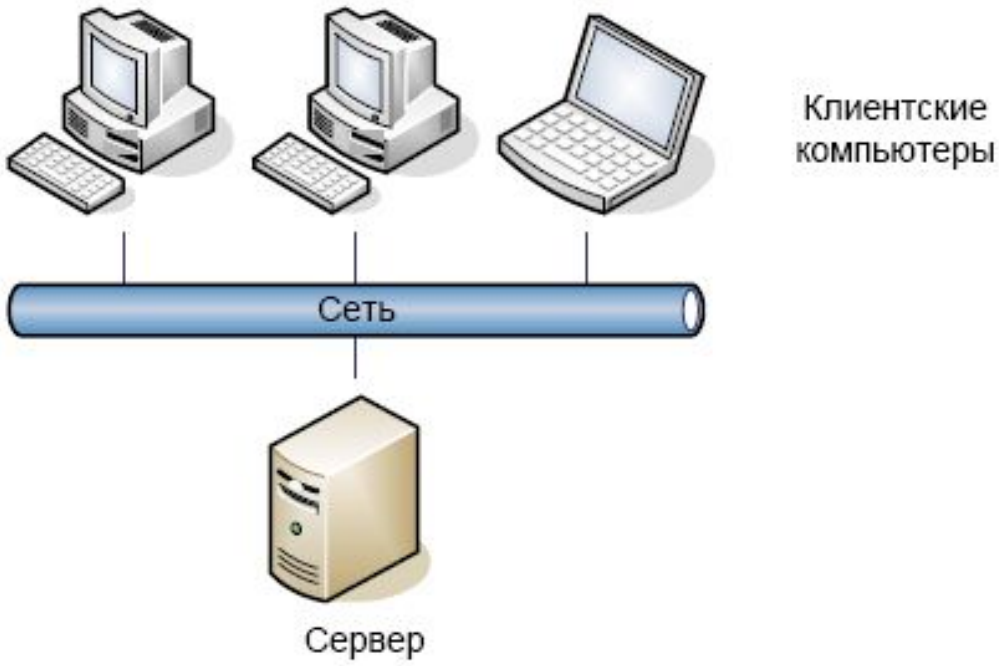
FUJITSU

Клиент-серверные приложения

- Архитектура «клиент-сервер» (2-tier applications)
- Трехуровневая архитектура «клиент-сервер» (3-tier applications)

Архитектура клиент-сервер

- Клиент
- Сервер БД (Oracle, MS SQL Server, DB2 и т.д.)



FUJITSU

Логическая структура приложений

Логические уровни:

- Пользовательский интерфейс
- Правила обработки информации (бизнес-правила)
- Управление данными

- *Если бизнес-правила реализуются на клиентской части приложения, то такого клиента называют «толстым»*
- *Если бизнес-правила реализуются на серверной части приложения, то такого клиента называют «тонким»*

FUJITSU

Преимущества и недостатки двухуровневой архитектуры

Преимущества:

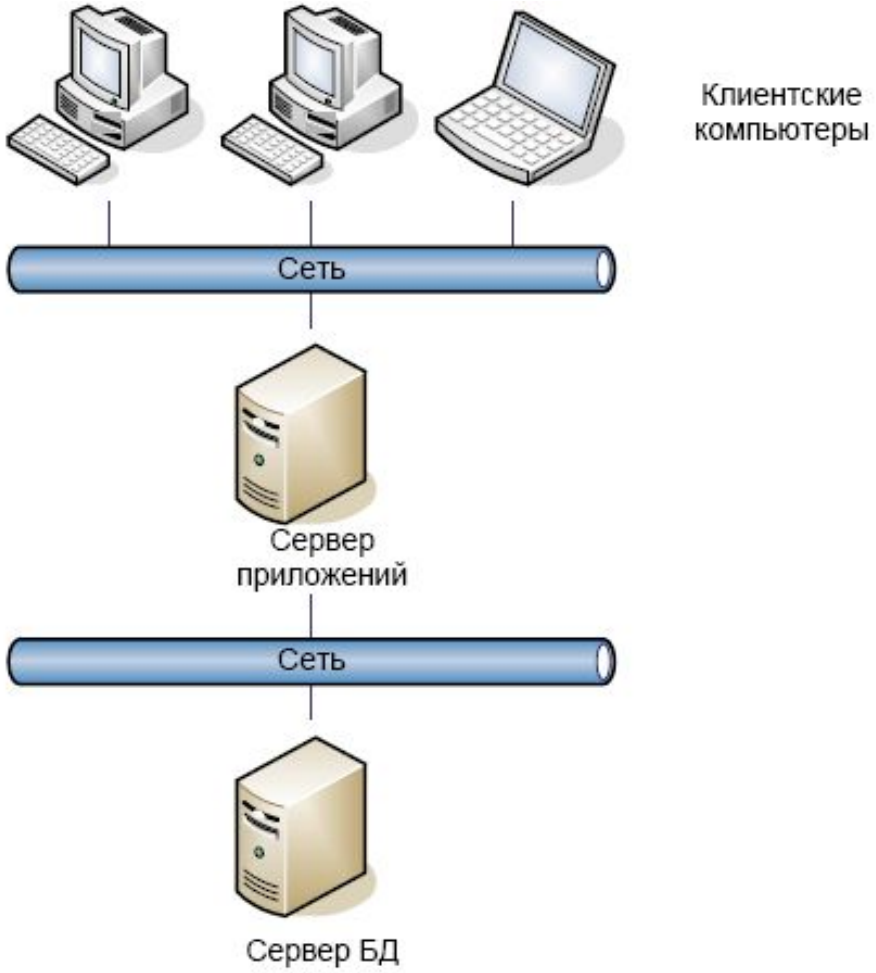
- Надежность
- Масштабируемость
- Безопасность
- Гибкость

Недостатки:

- Значительное увеличение загрузки сети при использовании «толстого» клиента
- Высокие требования к аппаратному обеспечению сервера при использовании «тонкого » клиента
- Необходимость установления соединения между базой данных и каждым работающим с системой пользователем, независимо от того, работает он в настоящий момент или нет

Трехуровневая архитектура

- Клиент
- Сервер приложений
- Сервер БД



FUJITSU

Преимущества и недостатки трехуровневой архитектуры

Преимущества:

- Высокая степень независимости от используемого сервера БД
- Снижение нагрузки «соединений»
- Простота администрирования системы

Недостатки:

- Наличие дополнительных «структурных элементов»
- Использование специальных механизмов обработки распределенных транзакций

FUJITSU

Элементы GUI windows-приложений

- Главное меню приложения (Main menu)
- Контекстное меню (Context menu)
- Список (List)
- Выпадающий список (Combo-box)
- Поля ввода
- Кнопка (Button)
- Переключатель (Check-box)
- Radio-button
- Таблицы
- Всплывающие подсказки

FUJITSU

Главное меню (main menu)

- Главное требование к меню – стандартизация. Цель стандартизации – облегчить пользователю работу с приложением.
- Размещение общих разделов приложения должно быть стандартизовано.
- Желательно, чтобы главное меню снабжалось инструментальной панелью.
- Расположение разделов в выпадающих меню должно быть стандартным
- Группы функционально связанных разделов отделяются в выпадающих меню разделителями

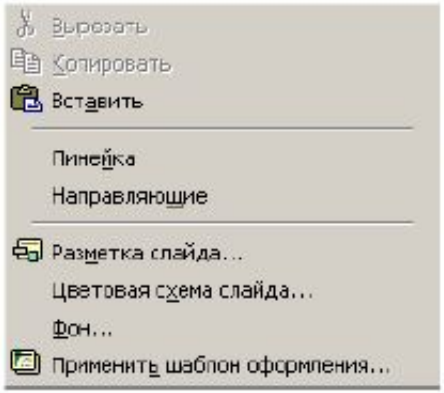
FUJITSU

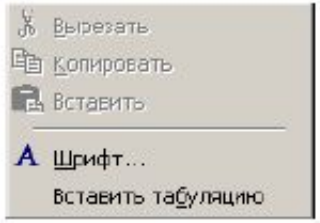
Главное меню (main menu)

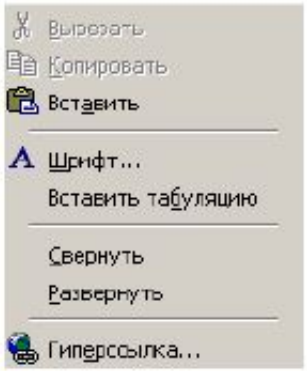
- В каждом названии раздела должен быть выделен подчеркиванием символ, соответствующий клавише быстрого доступа к разделу
- Многим разделам могут быть поставлены в соответствие “горячие” клавиши
- Не все разделы меню имеют смысл в любой момент работы пользователя с приложением. Такие меню и отдельные разделы должны делаться временно недоступными (Enabled) или невидимыми (Visible).
- Должно поддерживаться перемещение по пунктам меню с помощью кнопки tab и стрелок

Контекстное меню (context menu)

- Контекстное меню привязывается к конкретным GUI-компонентам
- Вызывается правым щелчком мыши, когда данный GUI-компонент находится в фокусе
- обычно в контекстное меню включают те команды главного меню, которые в первую очередь могут потребоваться при работе с данным компонентом.
- Некоторые приложения, устанавливаемые на компьютер, могут добавлять элементы в контекстное меню других приложений или даже в системное контекстное меню

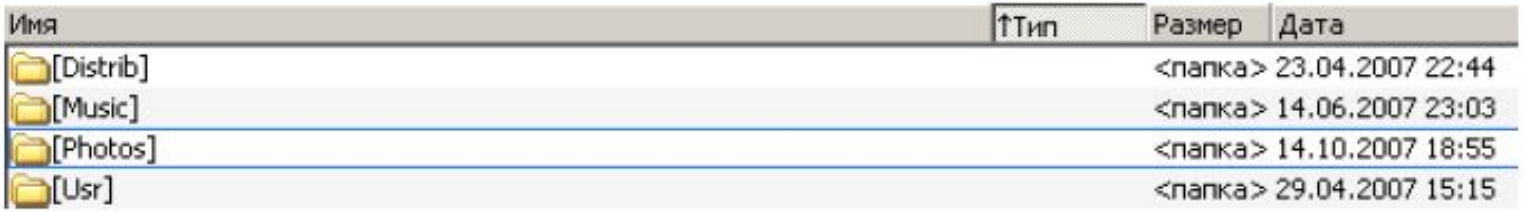






Список (List)

- Возможность сортировки по отдельным столбцам и/или группам столбцов
- Возможность выбора нескольких значений из списка
- Связанные списки:
  - при выборе значения из главного списка выполняется обновление привязанного списка
  - Перенос элементов между списками
- Список может быть «жестко» закодированным или же его значения загружаются из БД



Выпадающий список (combo box)

- Существует несколько типов выпадающих списков:
  - Список с фиксированным содержимым – позволяет выбирать только те значения, которые есть в списке
  - Список с фиксированными значениями и возможностью ввода – позволяет выбирать значения из списка либо указать свое значение
  - Пополняемый список – список позволяет выбирать значения из списка и вводить новые, причем новые значения добавляются в список
- Названия пунктов должны целиком умещаться в поле выпадающего списка
- В случае, когда значений в выпадающем списке очень много и все они не помещаются в окне компонента, должна присутствовать полоса прокрутки
- Выпадающий список может быть «жестко» закодированным или же его значения загружаются из БД
- Значения в выпадающих списках обычно сортируются

FUJITSU

Поля ввода

- Компонент предназначен для ввода и отображения информации
- Имеют ограничения длины вводимой строки
- Могут быть однострочными и многострочными
- Может не позволять вводить определенные символы (например, «@»)
- Должны поддерживать стандартные «горячие» клавиши работы с текстом (Ctrl+C, Ctrl+V, Ctrl+A, Ctrl+Z)
- Могут содержать элементы форматирования
- Поле для ввода пароля обычно имеет явное ограничение по длине (например, не более 8 символов) и отображает вводимые символы в виде «*»
- Поля ввода могут предусматривать «автозаполнение»

Поля ввода – ввод чисел и дат

- Поля для ввода чисел:
  - Обычно однострочные
  - Обычно имеют диапазон допустимых значений (например, только целые положительные числа)
  - Если поле для ввода чисел позволяет ввод дробных чисел, то в качестве разделителя дробной и целой частей может использоваться как «.», так и «,» - это может быть «жестко» закодировано или браться из настроек ОС. При использовании неправильного разделителя пользователь должен получать понятное уведомление, а не сообщение о неожиданной ошибке.
- Поля для ввода дат:
  - Однострочное поле
  - Позволяют вводить даты в определенном формате (например, dd.mm.yyyy)
  - Могут не позволять «ручной» ввод даты
  - Могут отображать календарь для выбора даты – следует проверять, какую дату по умолчанию предлагает календарь, есть ли возможность выбора текущей даты, возможность перелистывания календаря «по месяцам» и «по годам»
  - В зависимости от требований к приложению могут либо разрешать либо запрещать ввод даты в прошлом

FUJITSU

Поля ввода – набор проверок

- Проверить возможность ввода пустого поля
- Проверить ввод одного и нескольких символов
- Ввод пробельных символов в строке (в начале, в середине, в конце)
- Ввод строки, состоящей только из пробельных символов
- Ввод строки максимальной длины; строки, длина которой больше допустимой на 1 символ
- Ввод символов разного регистра
- Ввод специальных символов ' / ', ' \ ', ' ‘ ', ' “ ', ' # ', ' % ', ' & ', ' // ', ' \\ ', ' ~ ', ' ` ', ' @ ', ' $ ', ' ^ ', ' * ', ' (', ')‘, '?‘, '*'
- Ввод символов на языке, отличном от языка локализации ОС
- Проверка возможности работы с символами в поле (копирование, удаление и т.п.) с помощью стандартных комбинаций клавиш
- Выделение части строки с помощью «мыши» и комбинаций «shift+arrows»

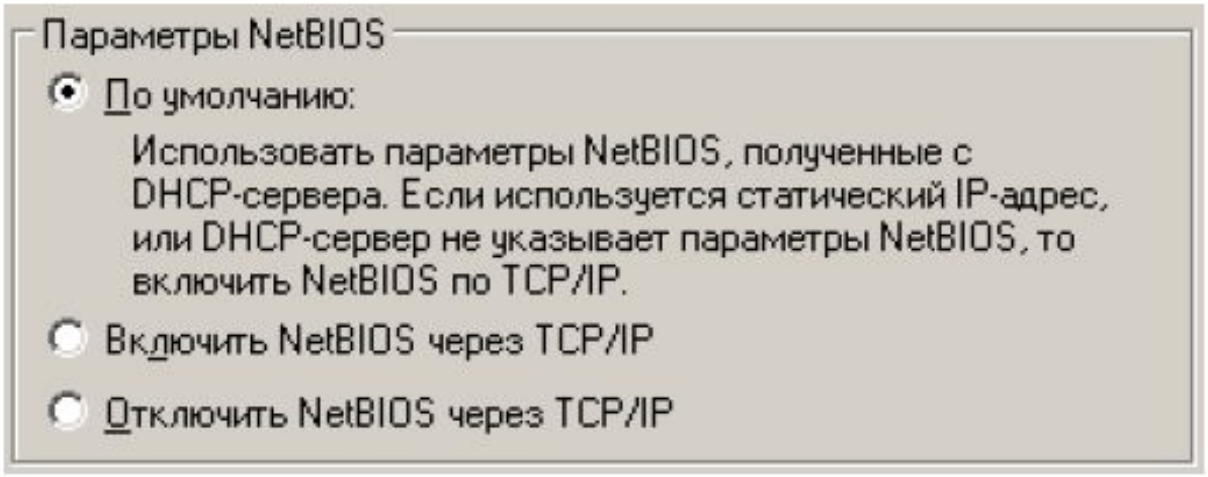
FUJITSU

Кнопка (Button)

- Позволяет выполнить какие-либо действия при нажатии кнопки во время выполнения программы
- Чек-лист:
  - Может быть доступной или недоступной для пользователя
  - Текст на кнопке должен лаконичным и понятным
  - Обычно кнопки в Windows-приложениях имеют прямоугольную форму
  - Кликабельный размер кнопки должен совпадать с ее видимым размером
  - Все кнопки в приложении должны быть выполнены в едином стиле
  - В качестве названия используется глагол, реже существительное
  - Нежелательно размещать кнопки вплотную друг к другу
  - При нажатии клавиши «пробел», если фокус установлен на кнопке, кнопка должна нажиматься
  - Наличие «горячих» клавиш для кнопки (например, для кнопки «Сохранить» комбинация «Ctrl+S»)

Radio-button

- Позволяет выбор только одного значения
- Обычно имеет общий заголовок
- Выбранное значение выделяется символом «точка»
- Обычно имеет значение «по умолчанию» (если это не так, то следует уточнить у разработчиков и аналитиков о том, правильно ли это)



FUJITSU

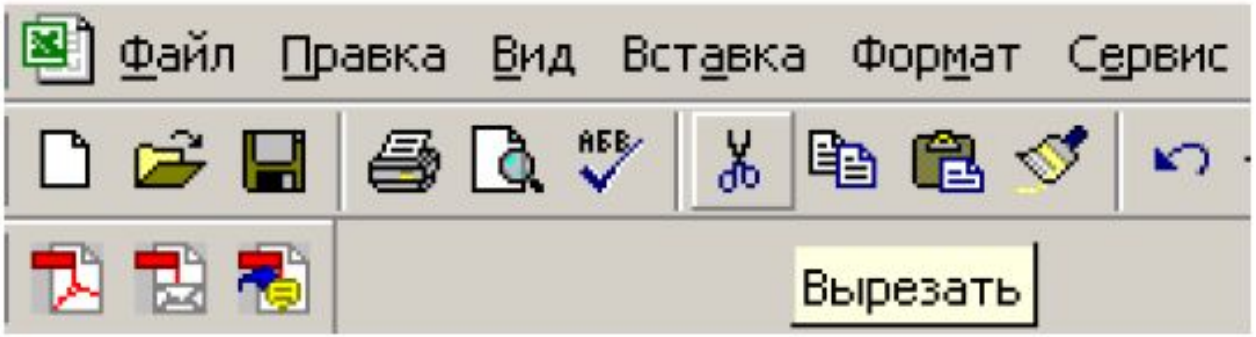
Таблицы

- Таблица – сложный элемент пользовательского интерфейса
- Количество строк и столбцов может быть как фиксированным, так и изменяемым (например, число строк может изменяться при добавлении записей – вводе новых значений)
- Если таблица большая и не помещается на экране, то должны присутствовать полосы прокрутки
- Обычно существует возможности сортировки по столбцам таблицы (также может быть предусмотрена сортировка по нескольким полям)
- В таблице может быть реализована функция поиска
- Ячейки таблицы фактически являются полями ввода, поэтому их следует проверять также как и поля ввода
- Если таблица размещается на нескольких страницах например при печати. нужно везде отображать заголовок
- Обычно таблицы можно экспортировать (например, в Excel-файл)
- Если таблицу можно напечатать, то необходимо проверять, что печатается только таблица, а остальные элементы интерфейса не печатаются, а также проверять, что таблица влезает по ширине
- Обычно, если в таблицу ставится значение, то в заголовке указывается единица измерения

FUJITSU

Всплывающие подсказки

- Всплывающие подсказки появляются при наведении курсора мыши на объект интерфейса пользователя
- Не являются обязательным требование, но являются признаком хорошего стиля
- Должны быть лаконичными, но в тоже время информативными



FUJITSU

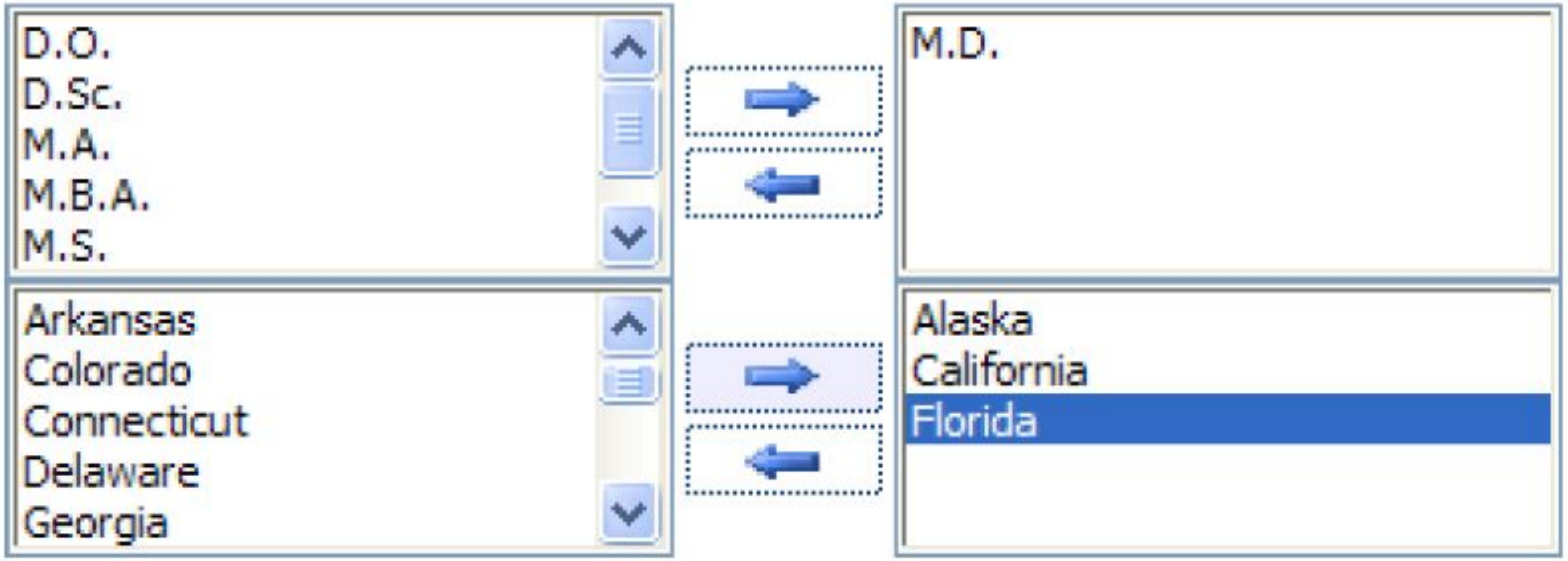
Сложные компоненты

- Современные приложения очень часто содержать компоненты, представляющие собой комбинацию нескольких элементов.
- Примеры: выпадающий список с возможностью редактирования, поля ввода с привязкой к БД, список с чек-боксами и пр.

- **Чем сложнее и функциональнее используемый компонент, тем больше вероятность того, что в нем содержатся ошибки.**
- **Чем больше времени разработчик потратил на дизайн (внешний вид, анимацию) компонента, тем вероятнее, что он допустил ошибку в функциональной части.**

FUJITSU

Сложные компоненты



FUJITSU

Инсталляционное тестирование

- Инсталляционное тестирование:
  - Тестирование процесса установки приложения (installation)
  - Тестирование процесса восстановления приложения (repair)
  - Тестирование процесса обновления приложения (upgrade)
  - Тестирование процесса удаления приложения (uninstall)
- Файлы конфигурации – назначение, форматы, тестирование

FUJITSU

Способы установки windows-приложений

- Способы установки Windows-приложений:
  - Установка из самораспаковывающегося архива
  - Использование «wizard»-инсталляторов
  - Установка приложений методом «copy-paste»
- Если приложению для корректной работы необходимо дополнительное ПО (например, виртуальная машина Java), то оно должно включаться в комплект установки (особенно важно для «коробочных» продуктов) или же в руководстве по установке должно быть явно прописано требование о предустановленном ПО
- В любом случае, если требуемое для установки/работы приложения ПО не установлено на компьютере, пользователь должен корректно информироваться об этом программой-инсталлятором во время начала установки приложения

FUJITSU

Возможные ошибки установки приложений

- Не работает установка приложения в папку отличную от значения по умолчанию
- Не создаются стандартные артефакты (ярлык на рабочем столе, пункт в меню «Пуск», отображение приложения в списке установленных программ и т.д.)
- Без уведомления пользователя переписываются системные файлы (например, dll), что может привести к ошибкам и даже к полной неработоспособности компьютера
- Не создаются или создаются неправильно записи в реестре Windows, переписывает существующие записи
- При установке приложения создается временная папка, которая после установки не удаляется
- Ошибки руководства по установке

FUJITSU

Тестирование восстановления приложений (repair)

- При восстановлении приложений должны копироваться и/или заменяться только те файлы, которые отсутствуют либо поврежденные файлы
- При восстановлении приложений все пользовательские настройки (в том числе настройки параметров подключения) не должны меняться, за исключением случаев, когда файлы конфигурации и настройки повреждены/удалены

FUJITSU

Тестирование обновления приложений (upgrade)

- При обновлении приложений все пользовательские настройки (в том числе настройки параметров подключения) не должны меняться
- Если после обновления приложения требуется перезагрузка, то следует проверить поведение приложения без перезагрузки – система должна корректно информировать пользователя о необходимости перезагрузки после обновления
- Должна существовать возможность отката обновление, за исключением случаев, когда это явно указано в требованиях

FUJITSU

Тестирование удаления приложений (uninstall)

- Способы удаления приложений:
  - Удаление с помощью «wizard»-деинсталляторов
  - Удаление приложений вручную
- Возможные ошибки:
  - При удалении программы могут удаляться системные dll и/или драйвера – это может привести к неработоспособности компьютера
  - При удалении программы могут без предупреждения (и без возможности выбора) удаляться виртуальные машины (Java, .Net)
  - При частичном удалении могут некорректно отрабатывать опции удаления
  - Некорректная «чистка» реестра Windows
  - Отсутствие операции «чистки» реестра Windows при удалении приложения тоже является ошибкой

FUJITSU

Конфигурационные файлы

- Конфигурационные файлы применяются для сохранения параметров, которые используются при работе программ.
- Форматы файлов конфигурации:
  - INI
  - XML
- Какие параметры обычно выносят в конфигурационный файл?
  - Каталоги (пути к лог-файлам, к файлам импорта/экспорта и пр.)
  - Сетевые настройки (имена серверов, IP-адреса и порты, имена и пароли для автоматического доступа и пр.)
  - Настройки БД (строки коннекта к БД и пр.)
  - Прочие (формат вывода дат и чисел и другие вещи, которые могут меняться от пользователя к пользователю)

FUJITSU

Конфигурационные файлы - тестирование

- При установке приложения необходимо проверять, какие значения записываются в файлы конфигурации (по умолчанию, в зависимости от опций установки)
- Приложение должно корректно обрабатывать неправильные значения и/или формат конфигурационных файлов и информировать об этом пользователя
- Проверка изменения конфигурационных файлов в процессе работы приложения (приложения требующие перезапуска, приложения обновляющие конфигурационные параметры по расписанию и/или по завершении работы – например, сохранение настроек, выполненных во время работы может также храниться в файлах конфигурации).
- Файлы конфигурации могут быть общими для всех пользователей, а могут быть отдельными для каждого пользователя – необходимо проверять, что те настройки, которые должны сохраняться в «личных» файлах, не попадают в «общие файлы».
- Распространенной ошибкой является «жесткое» кодирование значений параметров в приложении – т.е. несмотря на наличие параметра в конфигурационном файле, реальное значение прописано в коде программы, а не берется из файла.
- Хранение файла конфигурации на ресурсе, защищенном от записи – при попытке сохранения параметров система должна корректно информировать пользователя о невозможности записи.
- При обновлении или восстановлении приложения конфигурационные файлы не должны модифицироваться, если только это специально не оговорено.

FUJITSU

Влияние стороннего ПО

- Зависимость работоспособности Windows-приложений от внешних факторов:
  - Влияние антивирусов
  - Влияние брандмауэров (файерволов, сетевых экранов)
  - Влияние обновления операционных систем
  - Работа распределенного приложения при установке клиента и сервера в разных подсетях

FUJITSU

Тестирование совместимости

- Тестирование взаимодействия с внешними устройствами (принтеры, факсы, мобильные телефоны и т.д.):
  - Использование эмуляторов внешних устройств и контролеров внешних устройств (сложность – прежде чем использовать эмулятор необходимо провести его тестирование)
  - Тестирование при конкурентном использовании внешнего устройства
    - Функциональное тестирование в то время, когда устройство не занято
    - Отдельные тестовые испытания, проверяющие работу в режиме «конкуренции»
  - Проблемы с аппаратным обеспечением – драйверы устройств

Методика тестирования взаимодействия с внешними устройствами должна быть описана в плане и/или стратегии тестирования

FUJITSU

Сторонние компоненты

- Влияние сторонних компонентов (3-th party components)
  - Использование сторонних контролов в пользовательском интерфейсе приложения
  - Использование DLL-файлов, созданных сторонними разработчиками
- Ошибки, связанные с работой внешних приложений
  - Формирование отчетов в виде Excel-файлов
  - Показ рисунков с помощью предустановленных программ (например, ACDSee)
  - Работа с почтовыми клиентами

FUJITSU

Примеры ошибок

- В приложении разделителем дробной и целой частей чисел является запятая, а в ОС установлена точка. При формировании отчета в виде Excel-файла будет либо получена ошибка, либо отчет будет неправильно отображаться, если разработчик не предусмотрел функцию проверки.
- Приложение позволяет отправлять и получать почту, однако разработчик предусмотрел лишь возможность работы через MS Outlook. В случае, если в качестве почтового клиента Заказчик будет использовать другую программу или же почтовый клиент вообще не будет установлен, то приложение в лучшем случае не будет работать, а в худшем будет «падать».
- 30-го ноября протестировали обновление, отправили Заказчику. На завтра получаем сообщение от Заказчика – приложение падает. Снова тестируем у себя - у нас тоже падает. Понять ничего не можем. В конце выясняется, разработчик некорректно обрабатывает дату (строку в виде даты). Когда у него было 11/30/2005 - то все понятно, что формат mm/dd/yyyy. Но на следующий день приложение не могло "вычислить" используемый формат для даты, так как строка 01/12/2005 могла быть корректной датой и при mm/dd/yyyy и при dd/mm/yyyy. В результате приложение «падало», т.к. не могло определить формат даты.

Локализационное тестирование

- Локализационное тестирование – тестирование приложения, интерфейс которого реализован на различных языках (например, на русском, английском и французском):
  - Многоязычность приложения может быть реализована с помощью специальных файлов-справочников, либо с помощью специальных таблиц в БД.
  - Тестированием правильности перевода обычно занимаются переводчики
  - Задачи тестировщиков:
    - Проверить, что приложение правильно функционирует под всеми языками
    - При переводе слов на различные языки длины слов и сообщений могут меняться – необходимо проверить, что переведенные слова и сообщение не обрезаются и не выходят за границы видимых областей
    - Проверить печать в различных языках

FUJITSU

Ошибки в данных

- Неправильные данные как источник ошибок. Как избежать ошибок, связанных с неправильными данными:
  - Проверка импортируемых данных на соответствие формату и контенту приложения
  - Обработка некорректных данных при вводе

Очень важно понимать, что явилось причиной ошибки – неправильные данные или ошибка в логике.

FUJITSU

Тестовые данные

- Тестовые данные должны обеспечить проверку всех возможных условий возникновения ошибок:
  - Надо стремиться к тому, чтобы была испытана каждая ветвь алгоритма;
  - очередной тестовый прогон должен контролировать нечто такое, что еще не было проверено на предыдущих прогонах;
  - первый тест должен быть максимально прост, чтобы проверить, работает ли программа вообще;
  - арифметические операции в тестах должны предельно упрощаться для уменьшения объема вычислений;
  - количества элементов последовательностей, точность для итерационных вычислений, количество проходов цикла в тестовых примерах должны задаваться из соображений сокращения объема вычислений;
  - минимизация вычислений не должна снижать надежности контроля;
  - тестирование должно быть целенаправленным и систематизированным, так как случайный выбор исходных данных привел бы к трудностям в определении ручным способом ожидаемых результатов; кроме того, при случайном выборе тестовых данных могут оказаться непроверенными многие ситуации;
  - усложнение тестовых данных должно происходить постепенно

FUJITSU

Тестирование GUI

- Приложение должно корректно реагировать на изменение размеров окна:
  - Элементы не должны «накладываться» друг на друга - все части окна должны пропорционально изменятся
  - Элементы типа «таблица» должны либо «сжиматься», чтобы все столбцы помещались на новом размере окна, либо должна появляться полоса горизонтальной прокрутки
  - Если при изменении размера окна не все строки таблицы помещаются на экране, то должна появляться полоса вертикальной прокрутки
- Не все приложения позволяют изменять размер экрана

Модальные окна

- В графическом интерфейсе пользователя модальным называется окно, которое требует от пользователя завершения работы с ним до того, как будет возможно продолжить работу с родительским приложением. Модальные окна часто используются для привлечения внимания пользователя к важному событию или критической ситуации
  - При перемещении модального окна над родительской формой должна выполняться перерисовка родительской формы
  - Обычно модальные формы не позволяют изменять размер

FUJITSU

Usability testing

- Контрольный список проверок пользовательского интерфейса содержит ряд стандартных требований к пользовательскому интерфейсу:
  - Каждое окно приложения должно иметь заголовок
  - При нажатии на кнопку tab курсор должен переходить по полям ввода, в порядке соответствующем порядку заполнения полей
  - Кнопка Save должна находится ниже всех полей ввода или в левом нижнем углу
  - Нажатие на enter должно вызывать функцию сохранения
  - Элементы формы должны быть логично скомпонованы
  - Приложение должно отображать строку состояния (progress bar) при выполнении длительных операций
  - Должна быть возможность отмены действия и запрос на подтверждение рискованных операций (удаление)

FUJITSU

Тестирование безопасности

- Безопасность на уровне операционной системы
- Безопасность на сетевом уровне (шифраторы/дешифраторы, специальные средства проверки безопасности)
- Безопасность на уровне приложения

FUJITSU

Безопасность на уровне приложения

- Аутентификация на уровне приложения (требования к паролям)
- Разграничение прав доступа в соответствии с ролями
- Разграничение по наибольшему разрешению и по наибольшему запрету
- Хранение логинов и паролей в конфигурационных файлах и базе данных
- Прямой доступ к базе данных с помощью логинов пользователей приложения

FUJITSU

Приложения Java и .NET

- Особенности тестирования приложений Java и .Net:
  - При тестировании установки необходимо проверять установку приложения как на «чистый» компьютер (без установленных JVM и .Net framework), так и на компьютер, на котором уже установлены JVM и .Net framework.
  - При выходе новых версий JVM и .Net framework необходимо проверять работоспособность приложения.
  - Если приложение должно работать в разных ОС (например и в Windows и в Linux), то необходимо проводить тестирование во всех ОС.

Общие замечания

- Чем дольше работает ОС Windows, тем медленнее работают приложения и тем чаще возникают неожиданные ошибки.
  Возможные причины:
  - При установке очередного приложения изменились стандартные файлы библиотек (иногда даже на более старые версии)
  - При удалении приложения были удалены системные библиотеки
  - При установке/удалении программ вносятся изменения в реестр

Старайтесь определить, после чего у вас возникли ошибки. Как правило, на пустом месте они не происходят - этому предшествует либо установка какой-то программы, либо удаление.