Импортозамещение базовой ОС

Михаил АКСЕНОВ,
операционный директор,
компания «Национальная платформа»

Бруно ЧАВЕЗ
(Bruno Alejandro Sobieski Chavez),
технический директор,
компания «Национальная платформа»

Денис БЕЛОЦЕРКОВСКИЙ,
руководитель отдела разработки,
компания «Национальная платформа»

По нашему мнению, сейчас являются наиболее востребованными операционные системы на основе русской сборки ветки Debian Linux (например, Astra Linux), с доработкой значимого количества своих утилит и доработкой того, что позволяет сертифицироваться под требования ФСТЭК, МО и т.д. Например, мандатное разграничение доступа. Мы, как вендор российской ERP-платформы ориентируемся на наиболее популярные на рынке версии, а Debian одна из наиболее популярных веток Linux. Такие ОС являются современными, производительными системами. Они наследуют от исходной системы множество свойств, которые дополнятся собственными полезными разработками. Т.е. включают в себя как ноу-хау ОС Astra Linux, так и все накопленные возможности ОС, созданные всем сообществом, ведущим ветку Debian. В результате, если система работала ранее на Debian, необходимые усилия для ее адаптации, например, под ОС Astra Linux будут минимальными.

Особенности локализации ПО

Мы живем в глобальной деревне, с глобальным рынком, особенно в сегменте ИТ, где можем покупать/продавать программные продукты по всему миру, однако быть международным поставщиком программного обеспечения не так просто. Локализация ПО включает в себя не только перевод, но и ряд других требований разного уровня сложности, начиная с интернационализации программного обеспечения. Интернационализация – это процесс подготовки продукта к локализации. Среди основных характеристик интернационализированного программного обеспечения можно выделить следующие:

- разделение кода и содержимого;
- разделение текстов интерфейсов и их контекстуализация;
- поддержка различных форматов дат, валют, чисел и т. д.;
- расширение и сокращение текста и др.

Для успешного развертывания программы на российском рынке разработчику ПО необходимо учитывать ряд факторов, прежде всего связанных со сложностью языка, культурной адаптацией, соблюдением нормативных требований, техническими аспектами и локальной поддержкой.

Лингвистическая сложность

Важно обеспечить перевод не просто отдельных блоков текстов, но и каждого фрагмента с учетом общего контекста, в котором существуют текстовые блоки. Данное требование особенно актуально, когда целевым языком является русский – очень сложный язык с его грамматическими структурами и лексикой.

Лингвистические различия, такие как склонение, спряжение, наличие нескольких форм множественного числа и согласование полов, усложняют процесс локализации. Разработчики должны обеспечить точный перевод, чтобы сохранить удобство использования и понятность программного обеспечения.

Культурная адаптация

Локализация выходит за рамки перевода и включает в себя культурную адаптацию. Понимание и учет культурных норм, ценностей и предпочтений россиян крайне важны для создания удобного пользовательского интерфейса. Необходимо тщательно подбирать значки, символы, цвета и образы, которые находят отклик у российских пользователей. Адаптация форматов дат, символов валют, форматов адресов и систем измерения к местным традициям не менее важна для удобства использования программного обеспечения.

Несоблюдение культурных особенностей может привести к непониманию пользователей и отказу от применения программного обеспечения. Иными словами, недостаточно, чтобы программное обеспечение было корректным и выполняло поставленные задачи, оно должно подходить российскому пользователю.

Технические аспекты

Для российских операционных систем могут быть характерны уникальные технические требования, отличающиеся от других международных операционных систем. Это требует всестороннего тестирования для обеспечения совместимости, производительности и стабильности.

Уникальные технические требования включают в себя использование специальных сертифицированных библиотек и специфических протоколов шифрования. Если эти требования не будут соблюдены при разработке ПО, предназначенного для работы, например, в российском дистрибутиве Linux, то при развертывании в локальных российских средах программный продукт не будет иметь успеха.

Решение потенциальных проблем, связанных с кодировкой символов, поддержкой шрифтов, раскладкой клавиатуры, сертифицированными библиотеками и специфическими алгоритмами шифрования, критически важно для успешного процесса локализации программного обеспечения.

Локальная поддержка

Локализация программного обеспечения – это не только наличие готового к запуску продукта, но и обширная документация на местном языке. Кроме того, в зависимости от сложности программного обеспечения и целевого рынка требуется обеспечить поддержку на русском языке и адекватную систему контроля проблем.

Преодоление трудностей

Преодоление трудностей, связанных с локализацией программного обеспечения для российского рынка, зависит преимущественно от его направленности. Для потребительского ПО ключевыми факторами являются сотрудничество с носителями русского языка и российскими специалистами по интернационализации/локализации, а также наличие исчерпывающей документации.

В случае с бизнес-программами поставщикам ПО проще всего привлечь российские компании в качестве партнера, что не только обеспечит лингвистическую точность и культурную адекватность, но и поможет найти экспертов по местным нормам, требованиям безопасности и многим скрытым необходимым функциям, уже пройденным ими.

Наряду с этим интернационализация подготавливает бизнес-приложение к переносу на другие операционные системы. Отделение интерфейсных элементов от бизнес-логики помогает в переносе бизнес-приложения на другую ОС – можно отдельно выполнить перенос бизнес-логики и интерфейсных элементов. Интернационализированное приложение проще перенести на стороннюю операционную систему или на другой вариант того же Linux.

Портирование на отечественные ОС

Метод проверки совместимости приложения с новой операционной системой один: на целевой ОС – неважно, отечественной или зарубежной – запустить бизнес-приложение. Также необходимо провести исследование и определить необходимый минимум функциональности, который позволяет говорить о совместимости. Не менее важно зафиксировать это в рамках методологии, которую со временем надо расширять или/и пересматривать. В данном случае все зависит от бизнес-приложения. В одних случаях достаточно зафиксировать успешное открытие несколько раз подряд, в других – предстоит провести полноценное тестирование функционала (базового или с включением расширенного), а также нагрузочное тестирование. Требующееся на это время надо заложить перед выпуском очередной версии приложения. Хорошей практикой считается отслеживание изменений на целевых ОС или даже партнерские отношения с производителями ОС, чтобы получать доступы к ранним сборкам и иметь возможность проверить запуск своих приложений до официального выпуска на новой версии ОС.

В редких случаях может иметь значение и место, где установлена ОС: виртуальный сервер в облаке или физический в стойке, так как все зависит от уровня взаимодействия приложения с функциями операционной системы

и от того, насколько используются те или иные особенности ОС. Поэтому у качественного бизнес-приложения должно быть несколько вариантов сборок под разные виды ОС. Иными словами, бизнес-приложение должно быть кроссплатформенным. Каждая его версия проверяется на своем семействе ОС в соответствии с методологией.

В то же время варианты использования эмуляторов для запуска приложений на не поддерживаемой ранее операционной системе считаются плохой практикой для проверки совместимости. Например, WINE (хотя известно, что это не эмулятор – Wine Is Not an Emulator) обладает магией преобразования API-вызовов Windows в системные вызовы ядра Linux. Однако не всегда ясно, что получается на выходе. Работа приложения не гарантируется и может быть нарушена в непредсказуемом месте. Поэтому проекты типа WINE@Etersoft – скорее, шаг отчаяния, если вдруг приложение не кроссплатформенное и нет возможности иным образом запустить его на отечественной ОС. Не говоря уже о производительности и иных важных показателях.

От эмуляторов и преобразователей системных вызовов стоит избавиться, а еще лучше не начинать их использовать. Сегодня вместо них эксперты рекомендуют применять виртуальные машины или контейнеры с целевыми ОС.

При портировании бизнес-приложений на отечественные ОС следует соблюдать требования регуляторов. ФСТЭК России выдвигает требования по безопасности информации, устанавливая уровни доверия к средствам технической защиты информации и средствам обеспечения безопасности информационных технологий. Область применения данных требований – программное обеспечение, содержащее средства защиты информации и применяемое в информационных системах, обрабатывающих информацию ограниченного доступа, а также решения для обеспечения безопасности значимых объектов критической информационной инфраструктуры.

В большинстве случаев бизнес-приложения попадают в область соблюдения этих требований, так как обрабатываемая в них информация очень часто требует ограничения доступа и подразумевает установление собственных правил его разграничения. А это, в свою очередь, влечет за собой реализацию средств защиты информации как минимум в части идентификации/аутентификации пользователей, подсистем разграничения прав доступа к информации и аудита действий пользователей с информацией. В то же время требования ФСТЭК подразумевают использование компанией-разработчиком мер, направленных на предотвращение появления и устранение уязвимостей в программном обеспечении, которые могут быть использованы для нарушения конфиденциальности и целостности обрабатываемых данных.

Стоит заметить, что термин «доверенное ПО», введенный в широкую практику использования бизнес-сообществом в предложениях Минкомсвязи по импортозамещению программного обеспечения в 2014 г., несколько шире, чем безопасное программное обеспечение, как его рассматривает ФСТЭК России в сфере своей компетенции. Доверенное ПО аккумулирует не только требования ФСТЭК, но и требования по документированности бизнес-логики приложений, а также юридические аспекты – гарантии минимизации рисков отказа в обслуживании ПО со стороны компании-разработчика и минимизацию рисков, связанных с лицензионной чистотой продукта и его компонентов.

Таким образом, реализация требований к доверенному ПО требует от компании-разработчика полноценного внедрения стандартов промышленной разработки программного обеспечения, т. е. создания безопасного фундамента для реализации функциональных требований бизнеса. Таковыми являются система менеджмента информационной безопасности компании-разработчика; следование правилам разработки, оформления и обращения программ и программной документации; внедрение мер по разработке безопасного программного обеспечения (Secure Software Development Lifecycle).

Следовательно, отправной точкой создания бизнес-приложений, претендующих на использование в защищаемых информационных системах и обработку информации ограниченного доступа, становятся модели безопасности данных и модели угроз конфиденциальности, целостности информации, а во вторую очередь – функциональные требования к бизнес-приложению.

Заключение

Промышленный подход к созданию ПО перестал быть уделом «гаражного» программирования, поскольку требует сквозного документирования процессов разработки и управления изменениями ПО, а также взаимоувязанных правил разработки, оформления и обращения программ и программной документации. Помимо привычного функционального тестирования необходимо включение в процесс создания ПО действий для предотвращения появления и выявления уязвимостей, включая экспертизу, статический и динамический анализ исходного кода, фаззинг-тестирование и тестирование на проникновение.

Все это требует от разработчиков соответствующих инвестиций в создание бизнес-приложений, что особенно чувствительно для стартап-разработок. Существенную помощь оказывает практика государственного субсидирования программ импортозамещения ПО. Кроме того, появляются доверенные платформы разработки бизнес-приложений, которые реализуют подавляющее большинство требований к безопасному ПО и позволяют компаниям-разработчикам сфокусироваться на функциональных требованиях к своим бизнес-приложениям. ■