

Особенности локализации программного обеспечения на примере SCADA-системы WinCC

Рассматриваются современные методы региональной адаптации ПО, характерные проблемы и сложности этого процесса. Приводится обзор инструментальных программных средств локализации и практический опыт локализации SCADA-системы WinCC фирмы Siemens.

Необходимость региональной адаптации ПО

Все более глубокое проникновение ПК на все уровни иерархии современных промышленных предприятий приводит к вовлечению в работу с ними новых категорий пользователей. Можно выделить 2 наиболее характерные группы, находящиеся на разных социальных полюсах, которые были стремительно втянуты в среду новых информационных технологий, иногда даже помимо их собственного желания.

С одной стороны, это представители управленческого персонала – руководители предприятий и отделов, топ-менеджеры, главные инженеры проектов, а с другой – рядовые операторы, диспетчеры, просто рабочие. Несмотря на принципиальные отличия, обе эти группы представляют 2 важных сегмента потенциальных пользователей ПО, и успешное продвижение любого программного продукта на российских предприятиях возможно только в том случае, когда разработчики учитывают нужды и потребности обеих групп. “Управленцы” важны в силу их высокого положения и влияния на выбор закупаемого ПО, а с рядовыми пользователями нельзя не считаться как минимум по причине их многочисленности, а также из-за ответственности выполняемых ими действий по управлению ТП.

Предъявляя подчас диаметрально противоположные требования к ПО, представители обеих групп сходятся в обязательности наличия у любого приложения русскоязычного интерфейса. Для руководящего состава в большинстве случаев это лишь вопрос комфорта и удобства, в то время как для пользователей из числа второй группы интерфейс на иностранном языке может стать непреодолимым препятствием при работе с компьютерной системой. Перечисленные обстоятельства делают русскоязычный интерфейс не просто желательной особенностью, а абсолютно необходимой составляющей общей функциональности любой программной системы.

Кроме того, распространение нелокализованного ПО в некоторых странах запрещено законодательно, и российский рынок постепенно движется в этом же направлении [1]. Справедливое отчасти недоверие пользователей к русифицированным продуктам в 99 % случаев вызвано негативным опытом использования пиратских версий. Так, например, рунет изобилует возмущенными отзывами пользователей продуктов Adobe, варварски переведенных командами Alex Soft и D-Studio.

Особого рассмотрения заслуживают сложные программные комплексы (в частности, SCADA-системы), от которых требуется не только многоязыковая поддержка, но и возможность динамического переключения между несколькими языками. Сейчас уже и на российских предприятиях (особенно имеющих западных собственников или инвесторов) при визуализации ТП нередко предусматривают пользовательский интерфейс на нескольких языках, что дает возможность работы с системой иностранным наладчикам, менеджерам, аудиторам.

Побудительным мотивом к написанию данной статьи стала нелегкая, но плодотворная работа по адаптации SCADA-системы WinCC фирмы Siemens [2] для российских потребителей. Как выяснилось, многие интересные особенности региональной адаптации ПО в России известны только узкому кругу специалистов и крайне редко освещаются в технических и компьютерных изданиях. Хочется надеяться, что эта статья хотя бы отчасти восполнит данный пробел.

Процесс локализации и его основные этапы

В условиях доминирования западных производителей ПО большинство программных продуктов, представленных в сфере АСУТП, АСУП, MES- и ERP-систем, снабжаются русским интерфейсом не на стадии разработки, а лишь перед выходом на российский рынок. Осуществляемый при этом перевод пользовательского интерфейса программной системы и всех сопутствующих материалов на язык целевой аудитории называется *локализацией* (от английского – localisation и американского – localization, l10n)¹. *Локализация – это процесс адаптации ПО под конкретные национальные требования.* Локализации подвергается как заграничная продукция для продажи на внутреннем рынке, так и отечественное ПО перед продажей за границу. Не рассматривая в данной публикации особенностей локализации российских программ для иностранных потребителей, отошлем заинтересованных читателей к статье [3].

На самом деле локализация – это лишь часть процесса адаптации, которой всегда предшествует так называемая *глобализация* (от английского – globalization) или *интернационализация* (от английского – internationalization, i18n). Эти термины используются для обозначения способа проектирования ПО, при котором возможность многоязыковой поддержки закладывается с самого начала, а также процедуры модернизации уже существующего ПО с целью его аб-

¹ Разъяснения принятых в англоязычных источниках символично-численных обозначений i18n и l10n см. на <http://www.i18nguy.com/origini18n.html>

страгирования от особенностей языка и стандартов страны пользователя. В основу глобализации ПО положены две простые идеи, дающие возможность гибкой подстройки языкового окружения под конкретные требования без перекомпиляции приложений:

- хранение национально-зависимой части приложений отдельно от программного ядра в виде набора данных (так называемых объектов локализации);
- использование только предусмотренных в ОС стандартных методов представления и обработки информации, учитывающих все национальные особенности конкретных языковых сред.

В несколько упрощенном виде процедуру интернационализации можно представить в виде следующей последовательности действий:

- отделение всех переводимых элементов от программного ядра;
- обеспечение гибкости и языковой нейтральности пользовательского интерфейса;
- проверка поддержки необходимых наборов символов и поддержки требуемых национальных стандартов;
- обеспечение легкости перевода текста, включенного в растровые изображения.

Несмотря на концептуальную простоту, глобализация сопряжена со значительным числом технических сложностей, возникающих при ее практическом воплощении. Однако эта тема слишком обширна и представляет интерес только для профессиональных программистов, поэтому здесь мы ее касаться не будем. Заметим лишь, что методы и средства интернационализации и локализации ПО уже давно разработаны и закреплены на уровне стандартов корпораций-производителей ПО и независимых международных ассоциаций и консорциумов (например [4]), а неправильная работа программ в разных языковых средах почти всегда обусловлена нежеланием или неумением разработчиков эти методы и средства применять.

Локализация не ограничивается обычным переводом отдельных составляющих программного продукта – экранных форм, текстов сообщений, интерактивной справки и печатной документации. Собственно перевод – это трудоемкий, но далеко не единственный и не самый ответственный этап локализации. Помимо традиционного перевода всех исходных материалов требуется обеспечить корректность работы всей системы с новым интерфейсом и в новой языковой среде. В некоторых случаях этого можно достичь, лишь заново сформировав и протестировав продукт из переведенных компонентов. Вот лишь некоторые задачи, выполняемые в процессе локализации:

- перевод пользовательского интерфейса, включая изображения, голосовые сообщения и документацию;
- обеспечение работы системы с принятыми в регионе единицами измерения и стандартами;
- модификация фрагментов кода, обрабатывающих регионально-зависимые форматы данных (даты, время, номера телефонов и т.п.);
- обеспечение корректности лексикографической сортировки строк в соответствии с правилами целевого языка;
- настройка печати на стандартных для рассматриваемого региона форматах бумаги;
- проверка релевантности и допустимости в культуре целевой аудитории используемых символов, рисунков, цветовых комбинаций, музыкальных фрагментов и т.п.

Из приведенного перечня видно, что полная и качественная локализация достаточно крупного программного продукта – это сложный итеративный процесс, требующий участия специалистов различного профиля: технических переводчиков, составителей технической документации, разработчиков и испытателей ПО.

Типичные проблемы локализации

От качества локализации во многом зависит успешность продвижения продукта на национальном рынке. Крупные корпорации нередко тратят фантастические по российским меркам деньги на разработку пользовательского интерфейса программных систем. Это позволяет им получить качественный продукт, учитывающий особенности психологии, компьютерной грамотности, общего образовательного уровня, языкового багажа и культурных традиций целевой аудитории.

При этом результаты всех усилий разработчиков могут быть сведены на нет некачественной локализацией. Существует великое множество как объективных, так и субъективных причин, по которым локализованная версия ПО может быть существенно хуже оригинала¹. Ниже мы рассмотрим наиболее часто возникающие трудности локализации крупных программных пакетов, разделив их условно на технические, лингвистические, культурологические и организационные.

Технические аспекты

Наиболее безобидной и очевидной технической проблемой является часто встречающееся несовпадение длины текстов на языке оригинала и их переводов. К примеру, соотношение знаков при переводе с английского на русский в среднем составляет около 1:1,2, хотя в случае художественных переводов бывает и наоборот [5]. Это приводит к необходимости изменения размеров, расположения и выравнивания большинства интерфейсных элементов. К счастью, уже прошли времена “дикой” локализации, когда российским умельцам поручали взлом какого-либо популярного зарубежного продукта, а затем силами студентов-компьютерщиков, как-то владеющих английским языком, наспех и в условиях отсутствия необходимых исходных компонентов переводили наиболее важные части интерфейса.

¹ Верно и обратное: локализация иногда позволяет исправить ошибки оригинальной версии продукта.

Современные технологии локализации, как правило, не требуют от переводчика сохранения (не превышения) объема исходных текстов. Тем не менее, удлинение при переводе, например, лаконичного пункта меню “Rotate 90° CCW” на 25 символов до “Повернуть на 90° против часовой стрелки”, негативно сказывается на пользовательском интерфейсе системы.

Как это ни кажется странным, избыточная локализация почти всегда доставляет больше проблем, чем недостаточная. Эти проблемы касаются как пользователей, так и программистов, системных администраторов и специалистов служб технической поддержки. Оставляя в стороне уж совсем курьезные попытки полного перевода языков программирования (Ада, Кобол, Фокал) или создания русских (Ершол, Шпага, Рапира), отметим, что вопрос о том, что перевести, а что оставить “как есть”, актуален и сегодня. К примеру, переведя сообщение об ошибке загрузки шрифта для поддержки национального языка, локализатор почти наверняка лишает пользователя возможности его прочтения, в то время как в английском варианте это сообщение осталось бы читабельным.

Другой пример: имена файлов и каталогов. По большому счету пользователю должно быть безразлично, как именуются служебные объекты той или иной программной системы, но избавить его от необходимости выполнения действий на уровне файловой системы удается редко. В этом случае, даже если программная система поддерживает полное динамическое переключение языка пользовательского интерфейса, ни одна ОС не в состоянии “на лету” перевести имена файлов. Излишнее увлечение использованием национальных символов в именах файлов может доставить массу неприятностей и в современных ОС.

Например, в Windows 2000 с российскими региональными настройками файлы, содержащие в имени немецкий символ “ß”, некорректно обрабатываются стандартными командами ОС. Есть и другие примеры сложностей с поддержкой национальных языков в мире Windows. Так, специалисты знают, что в Windows 2000 продукты третьих фирм (например, Adobe Photoshop) и самой Microsoft (к примеру, SQL Server 2000) не работают с русским языком, если в системном реестре вручную не поправить ключ кодовой страницы 1252.

Более тонкие нюансы изменения поведения локализованного ПО подчас бывают скрыты не только от конечного пользователя, но и от переводчика. Если, скажем, разработчик приложения, взаимодействующего с Microsoft Word, при поиске открытых в системе документов ориентировался на название редактора в строке заголовка окна, а локализатор, проявив неумеренное рвение, перевел его как “Слово Microsoft”, приложение просто перестанет работать как надо.

Локализация программ, работающих с текстом, в ряде случаев требует добавления специализированных языковых модулей. Трудоемкость и сложность разработки таких модулей для русского языка, как правило, очень высоки в силу слабой формализуемости и нетривиальности алгоритмов обработки текстов. В этом случае не может не радовать то, что специалисты, занимающиеся адаптацией ПО для российского рынка, избавлены от целого букета специфических проблем, характерных для двухбайтовых (китайского, японского, корейского) и двунаправленных языков (иврит, арабский, фарси).

Лингвистические аспекты

Локализаторы ПО часто сталкиваются с проблемами, неведомыми традиционным переводчикам, главной из которых по праву можно признать отсутствие контекста. При переводе связанных текстов с английского на русский согласование родов, падежей и чисел не вызывает никаких осложнений. В то же время в программах диагностические сообщения, названия окон и содержимое строк состояния часто составляются динамически из заранее заданных словосочетаний и фрагментов предложений.

Таким образом, на первом этапе переводчик просто не в состоянии угадать, в каком контексте будет употребляться то или иное слово или фраза. В случаях, когда в программе один и тот же термин может переводиться на русский язык по-разному (picture – рисунок или кадр, line – линия или строка), а контекст отсутствует или недостаточен, локализатору остается уповать только на собственную интуицию и очень тщательное тестирование.

Часть подобных осложнений не возникает в тех случаях, когда система разрабатывается не на английском, и для программиста очевидно, что сказуемое может зависеть не только от числа (picture was updated/pictures were updated – кадр был обновлен/кадры были обновлены), но и от рода подлежащего (picture was deleted/library was deleted – кадр был удален/библиотека была удалена). Аналогичные неприятности вызывают и числительные. К примеру, англоговорящие разработчики выделяют только 2 случая – единственное и множественное число (1 page printed, X pages printed), в то время как в русском возможны варианты (распечатана 1 страница, распечатаны 2 страницы, распечатано 5 страниц), и в таких ситуациях приходится прибегать к различным уловкам (распечатано: X стр.).

Предложения с динамически подставляемыми параметрами вообще доставляют изрядные мучения локализаторам: в них нельзя менять порядок следования аргументов, даже если этого требует логика построения фразы. Ряд поучительных примеров неудачной англо-русской и русско-английской локализации приведен в статье [6].

Возникающие при локализации вопросы адекватности терминологии предметной области в целом характерны и для традиционного технического перевода, хотя неизбежное использование околомкомпьютерной лексики, не устоявшейся в русском языке, часто усложняет дело. Даже такому гиганту, как Microsoft, не удалось предложить достойный стандарт терминологии: в их международном списке слов [7] вы не найдете привычных “host”, “radio button”, “combo-box”, “spinner”, зато встретите совершенно неудобоваримые “указыватель” (pointer), “вешка разбивки” (split box), уж не говоря о банальных опечатках.

Шаткость общей компьютерной терминологии подчас усугубляется обилием “фирменной” терминологии разработчика, логика создания и использования которой доступна только самим создателям системы и очень опытным ее пользователям. Все это поистине превращает для переводчика поиск подходящего русского эквивалента в муки слова.

В области терминологии ПО имеется своя специфика, о существовании которой англоязычные ИТ-специалисты и лингвисты скорее всего не подозревают, и которую традиционно недооценивают носители русского языка. Для англоязычного читателя большинство научно-технических терминов – это немного переосмысленные слова родного языка, в то время как в России исторически сложилось, что родные слова с трудом воспринимаются как термины [8].

Хотя причина этого явления банальна и грустна – катастрофическое отставание во многих областях науки и технологий – эффект не столь однозначно негативен. За счет заимствований в русский язык вошло множество хорошо определенных точных понятий, не отягощенных сомнительными связями с исходными широкими значениями. Процесс этот, по всей видимости, естественный и неизбежный, хотя многие расценивают его как засорение русского языка.

Культурологические аспекты

Меньше всего сложностей при локализации ПО вызывают проблемы культурологического характера. Информационная среда современного мира в значительной мере является универсальной, абстрагированной от национальных, культурных и религиозных особенностей пользователей, и если обучающая программа для детей или экспертная система в процессе локализации могут претерпеть значительные изменения содержательного плана, то SCADA-система или пакет для программирования контроллеров вовсе не нуждаются в такого рода адаптации.

Тем не менее многие метафоры, используемые разработчиками программ, могут по-разному восприниматься представителями различных культур. Такая простая вещь как светофор может стать причиной недоразумений при переводе системы с одного языка на другой. В то время как в Англии, России и странах Европы используются красный, желтый и зеленый, в США – красный, оранжевый и зеленый, а в Японии, например, красный, оранжевый и голубой [1].

Всем привычный курсор в виде кисти с указующим перстом будет неадекватно воспринят в исламских странах, где запрещено изображать части человеческого тела. Даже мелочи, наподобие неблагозвучных или вызывающих негативные ассоциации торговых марок и аббревиатур, могут испортить впечатление от великолепного в остальных отношениях продукта.

Некоторые специалисты считают культурную адаптацию ПО на уровне пользовательского интерфейса принципиально недостаточной и доказывают необходимость модификации программного ядра систем [9]. При всей спорности такой точки зрения нельзя не признать существование практически неглобализируемых систем, изначально ориентированных только на региональных потребителей, например, 1С-Бухгалтерия, Гарант, Парус-Предприятие.

Организационные аспекты

Проблемы организационного характера, возникающие в процессе локализации ПО, типичны и предсказуемы. Это практически единственная сторона локализации, где можно научиться на чужих ошибках и сэкономить массу времени, сил и ресурсов, с самого начала правильно организовав работу. Основные сложности могут возникнуть, во-первых, во взаимодействии внутри команды локализаторов, и во вторых, – во взаимодействии локализаторов с разработчиками.

Внутренние проблемы локализаторов в основном сводятся к сложности поддержания единообразия терминологии и стиля программного продукта. Учитывая типичные сжатые сроки выполнения и высокую трудоемкость локализации, переводческую работу приходится распределять между группой исполнителей. Достижимое при этом ускорение отнюдь не линейно, так как возникают “накладные расходы” на выработку и согласование используемой лексики. Отметим также, что локализация ПО неанглоязычных компаний в силу экономической целесообразности часто проводится в 2 этапа.

Сначала выпускается англоязычная версия продукта, которая впоследствии уже переводится на остальные языки. Такой подход с одной стороны позволяет прибегать к услугам более широкого круга переводчиков с английского, но с другой – чреват копированием ошибок и недостатков английской версии. Российская локализация SCADA-системы WinCC, созданной немецкими программистами Siemens, как раз является примером двухступенчатой локализации, которая стала возможна благодаря наличию качественной английской версии.

В настоящее время разработка любого серьезного пакета ПО всегда сопровождается как минимум строгой стандартизацией терминологии, употребляющейся в пользовательском интерфейсе, а как максимум – составляется детальное руководство по стилю и оформлению, где регламентируются даже предпочтительные языковые обороты. Если разработчик предоставляет готовый глоссарий, все проблемы решаются его предварительным переводом и строгим контролем использования только рекомендованной лексики. В противном случае процесс перевода отдельных компонентов может стать итеративным, а по завершении все равно потребуются “сведение” терминологии и стилистики.

Специалисты, выполняющие локализацию, должны быть знакомы с основными канонами дизайна пользовательского интерфейса. К сожалению, в нашей стране их не всегда знают даже профессиональные разработчики программ, не говоря уж о переводчиках. Даже перевод Windows неидеален и в ряде случаев нарушает базовые правила организации человеко-машинного взаимодействия [10]. Вообще говоря, близкую к идеалу локализацию системы может провести только специалист:

- близко знакомый с предметной областью;
- в совершенстве владеющий языком, на который выполняется перевод (носитель языка);

- хорошо владеющий языком оригинала системы;
- являющийся опытным пользователем системы;
- обладающий широкой эрудицией в области программирования;
- имеющий опыт локализации программных систем.

Нехватка таких высококвалифицированных и универсальных специалистов часто приводит к необходимости создания тандема “профессиональный локализатор + эксперт по продукту и предметной области”.

Не меньшие сложности могут возникать при взаимодействии с заказчиком и разработчиком системы. Во-первых, не всегда создатели системы с готовностью предоставляют все необходимые исходные материалы, бережно охраняя свою интеллектуальную собственность. Во-вторых, выявляемые на этапе тестирования недочеты частично обусловлены неполной или некачественной интернационализацией продукта, исправление которой неизбежно требует вмешательства разработчиков и, как правило, срывает все сроки локализации.

В-третьих, для многоязычных систем добавление нового языка приводит не просто к появлению локализованной версии продукта, а к расширению и переработке основной версии, что опять-таки подразумевает тесное сотрудничество с разработчиками.

Жизнь заказчика тоже не безоблачна. Так как в общем случае его персонал не владеет целевым языком локализации, приходится либо доверяться исполнителю, либо искать независимых экспертов для оценки качества конечного продукта. Кроме того, однажды выпустив локализованную версию, производитель зачастую вынужден адаптировать и все последующие версии и пакеты обновлений.

При этом если права заказчика на все, в том числе и рабочие материалы локализации заранее не оговорены, он становится отчасти зависим от локализатора. Крупным иностранным компаниям приходится особо тщательно относиться к подбору исполнителей, в руках которых оказывается полнофункциональная версия коммерческого продукта: ни для кого не секрет, что в условиях российской действительности даже подписание соглашений о конфиденциальности не дает гарантий от появления пиратских копий.

Инструментарий для локализации

Хотя качество конечного продукта в решающей мере определяется профессионализмом, эрудицией, опытом и слаженностью команды, выполняющей региональную адаптацию, современный инструментарий помогает значительно упростить и ускорить процесс локализации, а главное – правильно его организовать. На сегодняшний день на рынке представлено несколько профессиональных систем локализации ПО. Среди наиболее известных можно перечислить Catalyst, Passolo, RC-WinTrans, Multilizer, Visual Localize. Практически все они имеют развитые средства обработки ресурсных данных, поддерживают проверку орфографии, ведение глоссария или терминологической БД, автоматизированное тестирование интерфейса, повторное использование имеющихся переводов, общую БД проекта для распределенной работы и множество других функций. Не имея возможности в рамках данной статьи провести подробный сравнительный анализ пакетов локализации, отошлем читателя к публикации [11], в которой эти пакеты рассмотрены достаточно детально и непредвзято.

Отличительными особенностями большинства пакетов локализации ПО являются достаточно высокая стоимость, защита аппаратным ключом и наличие нескольких версий продукта, отличающихся ценой и функциональностью. Например, компания Alchemy предлагает свой пакет Catalyst в вариантах Developer (€6500), Localizer (€4000) и Translator (€700), а также бесплатно предоставляет версию Lite, которая может работать только с проектами, созданными в редакции Developer. Другие производители используют похожие схемы продаж и лицензирования.

Анализируя собственный опыт интенсивной работы с системой Alchemy Catalyst v. 4.0 отметим, что пакет действительно удобный и по сравнению с обычным редактором ресурсов позволяет повысить эффективность и качество работы локализатора в несколько раз, если не на порядок. Однако мелкие недоработки, обходить которые удается очень замысловатыми методами, несколько портят изначально благоприятное впечатление от данного пакета.

Стоимость локализации

К локализации продукта следует с самого начала относиться как к неочевидной задаче, требующей творческого подхода, а также довольно больших временных и финансовых затрат [3]. Расценки на услуги локализации могут отличаться во много раз в зависимости от профессионализма исполнителей, языковой пары, объема и сложности продукта, качества предшествующей глобализации, особенностей предметной области и требуемых сроков выполнения работ. Расценки обычно приводятся в расчете на один символ или на одно слово исходного текста.

В качестве ориентировочной стоимости локализации специализированные компании указывают цифры около €0,025 за один символ, плюс разовая оплата ведения проекта, варьирующаяся от €50 до €300, хотя малоизвестные российские фирмы обычно работают по более низким расценкам. Часто предусматриваются скидки постоянным клиентам, планы сервисного обслуживания и другие гибкие схемы тарификации.

При поиске исполнителя следует обратить внимание на несколько нюансов. Во-первых, ни один грамотный локализатор никогда заочно не назовет стоимость работы. Заранее фиксированная ставка может означать либо некомпетентность исполнителя, либо то, что она получена в расчете на наихудший случай и заведомо невыгодна заказчику. Неадекватно низкие расценки также должны настораживать, т.к. никаких ноу-хау, позволяющих принципиально ускорить процесс перевода, не существует, труд локализатора – это “ручная работа”. Заранее следует согласовывать не

только расценки, но и метод подсчета объема исходных данных, а также возможные скидки при автоматическом переводе большого числа повторяющихся фрагментов.

Из опыта локализации WinCC

Вслед за версией WinCC, поддерживающей в стандартной конфигурации 5 европейских языков и дополнительно японский, китайский и корейский языки в азиатской поставке, фирма Siemens приняла решение произвести локализацию своей SCADA-системы для российского рынка. Как и любая SCADA-система, WinCC – достаточно большой и сложный программный продукт со своей спецификой, терминологией и техническими особенностями реализации. Не считая оперативной справки, объем текстов пользовательского интерфейса, подвергшийся локализации, составил приблизительно 150 тысяч слов. Весь цикл работ был условно разбит на 2 этапа: собственно локализация (2 месяца) и системный тест у заказчика (2 недели).

Перевод пользовательского интерфейса производился с использованием Alchemy Catalyst 4.0 редакций Developer/Pro и Quickship. В силу того, что WinCC предусматривает динамическое переключение языков как в среде исполнения, так и в среде разработки, помимо выполнения традиционных для локализации задач дополнительно пришлось изменять существующие и создавать новые фрагменты программного кода, обеспечивающие поддержку смены языка.

WinCC – ответственный программный продукт, от работы которого зависит огромное число пользователей по всему миру, в связи с чем выпуск пакетов обновлений и новых версий производится в строго оговоренные сроки. Российская локализация WinCC была проведена вскоре после выхода версии 6.0 SP1. До выхода очередной версии, в которую русский язык планируется включить наравне с 5 европейскими, было принято решение выпустить отдельный пакет поддержки русского языка, работающий со стандартной версией WinCC.

Об успешности локализации говорить пока преждевременно, об этом будут судить конечные пользователи, но предварительное тестирование несколькими компаниями-партнерами Siemens не выявило никаких серьезных недочетов.

Выводы

Подводя краткий итог вышесказанному, отметим, что локализация программного продукта важна в первую очередь как существенный фактор успешности его продвижения на региональном рынке. При этом лучше не проводить локализацию вовсе, чем выполнить ее некачественно. В этой связи особенно ответственно стоит подойти к выбору компании, выполняющей локализацию, т.к. успешность результата почти полностью зависит от опыта и квалификации ее сотрудников.

Лев Аронович Шерешевский – канд. техн. наук, главный специалист ООО “НВФ Сенсоры, Модули, Системы”.

Телефон (8462) 424-112.

E-mail: leo@industrialauto.ru

<http://www.sms-automation.ru>

Список литературы

1. Часто задаваемые вопросы по локализации ПО. Статья на сайте бюро научно-технического перевода (http://www.bntp.ru/human/faq_loc.asp).
2. Раздел Simatic WinCC на российском сайте департамента A&D фирмы Siemens (http://www.automation-drives.ru/as/products/simatic_hmi/wincc/index.php).
3. Анциферов А.А. Искусство локализации ПО или Не к ночи будет рассказано // Мир ПК. 1998. №10.
4. Сайт ассоциации по стандартизации индустрии локализации (LISA – Localization Industry Standards Association) (<http://www.lisa.org/>).
5. Михайлов М.Н. Чем длиннее, тем лучше? Как сравнить длины исходного текста и перевода? // Математическая морфология. Электронный математический и медико-биологический журнал. 2003, Т. 5, Вып. 1. (<http://www.smolensk.ru/user/sgma/MMORPH/TITL.HTM>).
6. Шахова Н. Локализация в свете глобализации // Компьютерра, 2001, №43 (420).
7. Международный список слов Microsoft (включен в библиотеку MSDN) (http://msdn.microsoft.com/library/en-us/dnwue/html/rus_word_list.htm).
8. Долгий А. Мини-словарь компьютерных терминов // Радио. 1999. №1.
9. Kersten G.E., Kersten M.A., Rakowski W.M. Software and Culture: Beyond the Internationalization of the Interface // Journal of Global Information Management, 2002, Vol. 10, No. 4.
10. Головач В.В. Дизайн пользовательского интерфейса. // Электронная книга. (<http://www.uibook1.ru/>).
11. Thomas Waßmer. Comparing Tools Used in Software Localization / MultiLingual Computing & Technology, 2002, No. 50, Vol. 13 Issue 6.