

УДК 004

МЕТОДОЛОГИЧЕСКИЕ АСПЕКТЫ ЛОКАЛИЗАЦИИ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ПРОГРАММНЫХ ПЛАТФОРМ АВТОМАТИЗАЦИИ БИЗНЕС-ПРОЦЕССОВ**Венюкова Е.А., магистрант**

Федеральное государственное автономное образовательное учреждение
высшего образования «Московский государственный технологический уни-
верситет «СТАНКИН»
г. Москва

Аннотация. Рассматриваются подходы к локализации пользовательских интерфейсов программных платформ автоматизации бизнес-процессов. Особое внимание уделено особенностям перевода динамических интерфейсов, работе с профессиональной терминологией и выбору инструментов локализации. На примере платформы п8п показано, какие трудности возникают при локализации сложных программных систем и как их можно решить. Отдельно рассматривается влияние локализации на эффективность работы пользователей.

Ключевые слова: локализация ПО, пользовательский интерфейс, интернационализация, п8п, автоматизация бизнес-процессов, перевод интерфейсов, САТ-системы.

Annotation. The structure of the client part of the open-source business process automation platform is analyzed using the example of the n8n system, internationalization mechanisms and the organization of language resources. It describes the implementation of working with text interface elements and what limitations arise when adding a new localization, taking into account the structure of localization files and the specifics of integrating language packs into the system. The results of the analysis make it possible to determine the key requirements for the localization architecture of complex software platforms and identify factors affecting the complexity of implementing a new interface language.

Keywords: software localization; internationalization; user interface; n8n; automation of business processes.

Автоматизация бизнес-процессов с появлением платформ по типу n8n такие инструменты стали доступнее, но появилась другая проблема, заключающаяся в языковом барьере. На практике это выглядит довольно неудобно, потому что вроде бы интерфейс есть, функциональность есть, но часть пользователей тратит время не на работу, а на попытки понять, что именно означает тот или иной элемент. В результате снижается скорость работы и увеличивается количество ошибок. Поэтому локализация интерфейса актуальна и востребована.

Поскольку текстовые элементы содержатся в коде, то любой перевод превращается в длительный процесс. В правильно спроектированной системе текст вынесен в отдельные ресурсы, например, JSON-файлы, а интерфейс получает нужные строки по ключам. В случае с n8n реализована как раз именно такая архитектура, когда интерфейс использует систему ключей и языковых файлов, что позволяет добавлять новые языки без изменения бизнес-логики.

Перевод интерфейса сложной системы заметно отличается от перевода сайта. Выражается это в том, что интерфейс платформы динамический. Один и тот же элемент может отображаться в разных контекстах, и перевод должен учитывать это. Без понимания логики системы легко получить формально правильный, но фактически неверный перевод.

Система также содержит большое количество коротких строк. Кнопки, подписи, уведомления часто содержат 1–2 слова без контекста и перевод «в вакууме» здесь не работает.

Еще одним из нюансов локализации является то, что пользователи n8n – это технические специалисты: аналитики, разработчики, маркетологи. У них уже есть сложившийся словарь, и попытка перевести всё на русский язык иногда только ухудшает восприятие, потому что, например, такие термины, как «workflow» или «trigger», в ряде случаев логичнее оставить без перевода или использовать гибридные варианты [1].

Основные трудности возникают не из-за языка как такового, а из-за его использования в интерфейсе [2]. На экран выводятся плейсхолдеры и динамические строки. Такие фразы с переменными значениями требуют перестройки предложения и простой прямой перевод часто приводит к грамматическим ошибкам.

Русский язык сложнее английского в плане склонений и числительных. Это особенно заметно в уведомлениях. А также интерфейс накладывает жесткие ограничения, ведь иногда приходится сокращать текст, не теряя смысла.

Без понимания, где используется строка, перевод может быть неверным. Поэтому в реальных проектах переводчик должен работать с интерфейсом, а не только с текстом. Полностью автоматический перевод в таких задачах практически бесполезен. Он дает быстрый, но поверхностный результат.

На практике лучше работают САТ-системы. Они позволяют сохранять ранее переведенные сегменты, использовать глоссарии, отслеживать одинаковые формулировки и проверять корректность перевода [3].

В рассматриваемом проекте использовалась система MateCat, которая сочетает машинный перевод и память переводов. Это не заменяет переводчика, но заметно снижает количество повторяющейся работы.

На практике этот процесс редко выглядит как аккуратная последовательность шагов. Скорее, это несколько взаимосвязанных задач, к которым приходится возвращаться снова и снова.

Работа обычно начинается с попытки разобраться в архитектуре приложения. Без этого перевод превращается в угадывание. Нужно понять, где именно лежат текстовые ресурсы, как они подставляются в интерфейс и какие части системы вообще участвуют в отображении текста. Иногда это занимает больше времени, чем сам перевод, особенно если структура проекта запутанная или документация неполная [4].

Параллельно постепенно формируется глоссарий. Причем не как отдельный документ «на потом», а прямо по ходу работы. Один и тот же термин может встретиться десятки раз в разных местах интерфейса, и, если каждый раз переводить его по-разному, это сразу бросается в глаза. В какой-то момент начинаешь ловить себя на том, что проверяешь не перевод, а согласованность формулировок.

Сам перевод почти всегда идет с привязкой к контексту. Текст сам по себе мало что значит, особенно если это короткие подписи или кнопки. Приходится постоянно сверяться с интерфейсом и смотреть, где появляется строка, что происходит после нажатия, какие действия ожидает пользователь. Без этого легко сделать «правильный» перевод, который в реальности звучит странно или сбивает с толку.

Отдельный пласт работы состоит из динамических строк. Уведомления, сообщения об ошибках, текст с подставляемыми значениями. Здесь уже недостаточно просто перевести слова и нужно учитывать, как будет выглядеть фраза после подстановки чисел или параметров. Иногда проще полностью перестроить предложение, чем пытаться сохранить структуру оригинала [5].

Также часто недооценивают заключается в тестировании с пользователями. Именно здесь всплывают вещи, которые невозможно заметить в одиночку. Формулировка может казаться понятной, пока не увидишь, как другой человек пытается интерпретировать ее в реальной задаче. Такие мелкие недочеты быстро накапливаются и начинают влиять на общее восприятие системы. Такой подход позволяет избежать типичной ситуации, когда перевод формально завершен, но пользоваться интерфейсом неудобно [6].

На рисунке 1 представлена последовательность основных этапов локализации пользовательского интерфейса программной платформы. Схема отражает переход от анализа архитектуры приложения и выделения строковых ресурсов к формированию терминологической базы, переводу с учетом контекста и последующему тестированию.



Рисунок 1. Последовательность этапов локализации интерфейсов программной платформы

Опыт работы с платформой n8n показал, что основные сложности связаны не с технической реализацией, а с выбором подхода к переводу, где нужно учитывать контекст, терминологию и особенности аудитории.

При правильной организации процесса локализация дает измеримый эффект и сокращает время выполнения задач и снижает количество ошибок.

Список литературы

1. Antwiadjei L., Huma Z. Comparative Analysis of Low-Code Platforms in Automating Business Processes // Asian Journal of Multidisciplinary Research & Review. – 2022. – Vol. 3. – No. 5. – P. 132–139.
2. Легенько А. Н. Языковая локализация как вид переводческой деятельности // Вестник науки. – 2023. – Т. 3. – № 5 (62). – С. 553–562.
3. Генке С. В. Обзор работы САТ-систем. Что это и зачем необходимо переводчику // Вестник науки. – 2025. – № 6 (87).
4. Лиманова Н. И., Селезнев И. А. Анализ эффективности клиент-серверной архитектуры // Бюллетень науки и практики. – 2022. – № 7.
5. Худенко Е. Перевод и локализация: введение в профессию. Основы, советы, практика. – Москва : Издательские решения, 2021. – 200 с.
6. Roturier J. Localizing Apps: A Practical Guide for Translators and Translation Students. – New York : Routledge, 2015. – 192 p.