РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРНЫЙ (АРКТИЧЕСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ М.В. ЛОМОНОСОВА»

На правах рукописи

ВОВЧЕНКО Вячеслав Евгеньевич

ПЕРЕВОДЧЕСКИЕ НЕУДАЧИ В ПЕРЕВОДЕ АНГЛОЯЗЫЧНОЙ ДОКУМЕНТАЦИИ ПО ВЕБ-РАЗРАБОТКЕ НА РУССКИЙ ЯЗЫК

Специальность 5.9.8. Теоретическая, прикладная
и сравнительно-сопоставительная лингвистика

ДИССЕРТАЦИЯ
на соискание ученой степени
кандидата филологических наук

Научный руководитель –
доктор филологических наук,
профессор
Поликарпов Александр Михайлович

Архангельск – 2026

ОГЛАВЛЕНИЕ

Введение 4
1. Трансфер знаний в ИТ-сфере и техническая документация как важный его элемент 17
1.1. Трансфер знаний и его роль в развитии информационных технологий 17
1.2. Роль веб-разработки как области информационных технологий в трансфере знаний 25
1.3. Роль трансфера знаний в развитии индустрии перевода 31
1.4. Место веб-разработки в современной ИТ-индустрии 37
1.4.1. Техническая документация как разновидность технических текстов 45
1.4.2. Особенности перевода текстов документации в области веб-разработки 59
1.5. Переводческие ошибки и неудачи в контексте перевода технической документации 68
1.5.1. Переводческая ошибка как категория переводоведения 69
1.5.2. Классификации переводческих ошибок: обзор 76
1.5.3. Понятие переводческой неудачи и его методологический потенциал 81
1.5.4. Переводческое стратегирование 86
1.5.5. Типология переводческих неудач 92
Выводы по главе 1 100
2. Документация к ключевым технологиям веб-разработки: React JS, Vue.js и MDN Web Docs 104
2.1. React JS: краткая история развития, значимость в веб-разработке и структура документации 104
2.2. Vue.js: краткая история развития, значимость в веб-разработке и структура документации 111
2.3. MDN Web Docs: краткая история развития, значимость в веб-разработке и структура документации 116
Выводы по главе 2 121

3. Анализ переводческих неудач в переводе англоязычной документации по веб-разработке на русский язык 125
3.1. Материал и методика исследования 125
3.2. Лексико-семантические переводческие неудачи 130
3.2.1. Статистическая характеристика и общая типология 130
3.2.2. Лексико-семантические переводческие неудачи при передаче метаязыка документации 132
3.2.3. Лексико-семантические переводческие неудачи, связанные с терминологическим обеспечением согласованного употребления терминов 134
3.3. Прагматические переводческие неудачи 143
3.3.1. Статистическая характеристика и общая типология 144
3.3.2. Прагматические неудачи, связанные с модальной семантикой 146
3.3.3. Прагматические неудачи, обусловленные искажением директивности и инструктивности 152
3.3.4. Прагматические неудачи, связанные с утратой эмоционального тона 156
3.3.5. Прагматические неудачи, затрагивающие невербальные элементы документации 159
3.4. Стилистические переводческие неудачи 168
3.4.1. Статистическая характеристика и общая типология 168
3.4.2. Стилистическая перегруженность и канцеляризация 171
3.4.3. Стилистическая сниженность и разговорность 177
3.5. Переводческие неудачи синтаксического характера 183
3.6. Гибридные переводческие неудачи и их кумулятивный эффект 191
Выводы по главе 3 196
Заключение 201
Список литературы 207

Введение

В условиях стремительного развития цифровых технологий всё более заметную роль в профессиональной коммуникации играет качественная передача специальных знаний. Особенно отчётливо это проявляется в сфере информационных технологий, где освоение новых инструментов во многом зависит от того, насколько последовательно, ясно и точно представлена сопровождающая их документация. Для значительной части русскоязычных специалистов переводная документация становится одним из основных источников знакомства с новыми технологиями веб-разработки, их принципами работы и рекомендуемыми практиками использования. В связи с этим перевод в данной области приобретает особую значимость, поскольку от него зависит доступность профессионального знания и характер его дальнейшего усвоения.

Тексты документации по веб-разработке обладают рядом особенностей, которые делают их привлекательным объектом для научных исследований. Они сочетают свойства технического и учебного текста, включают в себя терминологически насыщенные объяснения, пошаговые инструкции, элементы гипертекста, примеры кода и комментарии к ним, что требует от переводчика высокой степени внимательности в передаче смысла, терминологии, логики изложения и прагматической направленности текста. Изучение переводческих неудач в англоязычной документации по веб-разработке и её переводах на русский язык позволяет понять специфику перевода данной разновидности текстов и выявить факторы, влияющие на качество трансфера специальных знаний в области веб-разработки в русскоязычную профессиональную среду.

Степень разработанности темы. В современной отечественной и зарубежной лингвистике и переводоведении накоплен значительный опыт, связанный с

изучением технических текстов, специальной терминологии, научно-технического перевода, перевода текстов в сфере информационных технологий, а также с вопросами качества перевода и функциональной адаптации специального текста. Отдельные аспекты, имеющие значение для настоящего исследования, рассматриваются в научных трудах, посвящённых изучению различных аспектов технической документации, ИТ-дискурсу и трансферу специальных знаний в ИТ-секторе. Вместе с тем документация по веб-разработке как самостоятельный объект исследования до настоящего времени с лингвистических и переводческих позиций практически не изучалась. В существующих исследованиях она, как правило, не отделяется от более широкого массива технических или ИТ-текстов, вследствие чего её жанровая, прагматическая и когнитивная специфика остаётся недостаточно описанной. В особенности это касается перевода англоязычной документации по веб-разработке на русский язык и тех случаев, когда перевод сохраняет предметное содержание оригинала, но приводит к ослаблению инструктивности, нарушению терминологической последовательности, изменению характера подачи материала или снижению его обучающего потенциала.

В последние годы всё более заметным становится интерес к тем видам профессиональной коммуникации, в которых язык участвует в передаче, структурировании и освоении специального знания. К таким видам текста относится и документация по веб-разработке, сопровождающая все современные веб-технологии. Она широко используется в обучении, самообучении и практической деятельности веб-разработчиков, а качество её перевода напрямую влияет на степень понимания технологических концепций, последовательность усвоения терминологии и корректность дальнейшего применения описываемых решений. Рассмотрение документации по веб-разработке и её перевода представляется особенно значимым в русле интегративного подхода к переводу, позволяющего учитывать языковые, речевые, коммуникативные, когнитивные, семиотические, производственно-экономические и некоторые другие аспекты перевода. **Актуальность** данного

исследования определяется важной ролью англоязычной документации по веб-разработке в современной цифровой среде, растущей потребностью в её качественном переводе на русский язык и повышением интереса учёных к изучению трансфера специальных знаний и к переводу как системной деятельности, способствующей передаче и адаптации информации без существенных потерь, а также к исследованию явлений, препятствующих этому с позиций оценки качества перевода.

Объектом исследования являются англоязычные оригиналы и русскоязычные переводы официальной документации, содержащие описания современных веб-технологий.

Предметом исследования являются переводческие неудачи, возникающие при переводе англоязычной документации по веб-разработке на русский язык.

Цель исследования заключается в выявлении и описании переводческих неудач, наблюдаемых в переводе англоязычной документации по веб-разработке на русский язык.

Данная цель обусловила постановку следующих задач:

1) определить теоретические подходы к изучению трансфера знаний в сфере информационных технологий и выявить роль документации по веб-разработке в данном процессе;

2) охарактеризовать документацию по веб-разработке как особую разновидность технического текста и установить её функциональные, содержательные и структурные характеристики, значимые для оценки качества перевода;

3) обосновать возможность использования интегративного подхода к рассмотрению перевода документации по веб-разработке и уточнить в связи с изучением проблем оценки качества перевода эффективность обращения к понятиям переводческой неудачи и переводческой ошибки и их размежевания применительно к изучаемой разновидности интернет-текста;

4) охарактеризовать и описать официальную документацию к технологиям React JS, Vue.js и MDN Web Docs с целью представления исследовательского корпуса для последующего изучения переводческих неудач;

5) выявить и систематизировать на отобранном языковом материале переводческие неудачи, обнаруживаемые в переводе англоязычной документации по веб-разработке на русский язык;

6) описать переводческие неудачи лексико-семантического, прагматического, стилистического и синтаксического характера в переводах документации по веб-разработке с учётом интегративного подхода к их рассмотрению и трансфера специального знания, осуществлению которого они в значительной мере препятствуют;

7) определить влияние выявленных переводческих неудач на функциональную состоятельность перевода документации по веб-разработке как средства трансфера специальных знаний.

Методы исследования определялись в соответствии с поставленной целью и задачами. В ходе работы использовались метод лингвистического описания, включающий наблюдение, интерпретацию, обобщение и систематизацию материала; метод сопоставительного анализа, применяемый при сравнении англоязычных оригиналов и русскоязычных переводов документации; лексико-семантический анализ, направленный на выявление переводческих неудач в передаче терминов и метаязыка документации; прагматический анализ, позволяющий рассмотреть изменения в коммуникативной направленности в результате перевода, инструктивности и модальности текста перевода в сопоставлении с текстом оригинала; стилистический анализ, используемый для выявления жанрово-стилистических параметров оригинала и перевода; синтаксический анализ, помогающий регистрировать переводческие неудачи, влияющие на структурное оформление высказывания в переводе; элементы дискурсивного анализа, используемые для учета функциональной специфики документации по веб-

разработке; метод классификации, применяемый для систематизации выявляемых переводческих неудач, а также метод количественных подсчётов, используемый для установления общего количества выявленных переводческих неудач, определения их распределения по источникам материала и типам, а также для сопоставления частотности разных групп неудач в исследуемой документации.

Материалом для исследования послужили англоязычные оригиналы и русскоязычные переводы официальной документации к ключевым технологиям веб-разработки: React JS, Vue.js и MDN Web Docs. В качестве источника англоязычного материала использовались официальные версии документации, размещённые на сайтах legacy.reactjs.org, vuejs.org и developer.mozilla.org, а в качестве переводного материала – соответствующие русскоязычные версии этих ресурсов. Исследованию подверглись микроконтексты, извлечённые из обучающих, справочных, инструктивных и пояснительных разделов документации, а также заголовки, комментарии к примерам кода, фрагменты гипертекста и иные элементы текста, значимые для оценки качества перевода.

Гипотеза исследования основывается на предположении о том, что при переводе англоязычной документации по веб-разработке на русский язык можно обнаружить переводческие неудачи, препятствующие осуществлению трансфера специального знания и в значительной степени снижающие качество перевода, что влияет на обеспечение устойчивости цифровой среды и в конечном итоге приводит к локальным и глобальным сбоям. Предполагается, что такие неудачи могут иметь системный характер, затрагивать разные аспекты организации документации и связаны с недостаточностью и непоследовательностью переводческого стратегирования, вследствие чего перевод в той или иной мере утрачивает функциональную состоятельность как средство структурированного трансфера специальных знаний.

На защиту выносятся следующие положения:

1) Документация по веб-разработке представляет собой особую разновидность специального текста, в котором сочетаются признаки технического и учебного текстов. Перевод данной документации должен оцениваться с учётом эквивалентности в передаче терминов и элементов метаязыка документации, точности в передаче предметного содержания, адекватности в отношении выполнения текстом обучающей и инструктивной функций и «читабельности» текста перевода.

2) Применительно к переводу англоязычной документации по веб-разработке понятие переводческой неудачи обладает большей объяснительной силой, чем понятие переводческой ошибки в её традиционном понимании. Переводческие неудачи рассматриваются как отклонения от стратегического планирования, которые связаны со снижением коммуникативной эффективности перевода, ослаблением прагматического потенциала, в том числе обучающего, изменением исходной тональности, нарушением жанровых ожиданий адресата и, как следствие, частичной утратой функциональной состоятельности документации.

3) Интегративный подход к оценке качества перевода создаёт эффективную методологическую основу для изучения переводческих неудач в документации по веб-разработке, поскольку позволяет учитывать языковые, речевые, коммуникативные, когнитивные, семиотические, производственно-экономические и другие аспекты перевода. В рамках данного подхода перевод рассматривается как комплексный процесс, ориентированный на достижение функционально значимого результата в конкретной профессиональной среде. Это делает возможным описание переводческих неудач как проявлений рассогласования между исходной функцией документации и её переводом.

4) Переводческие неудачи в англоязычной документации по веб-разработке на русский язык могут иметь системный характер и проявляются в различных аспектах текстовой организации. Они обнаруживаются в лексико-семантическом,

прагматическом, стилистическом и синтаксическом аспектах, а в ряде случаев имеют гибридный характер.

5) В лексико-семантическом аспекте наибольшее значение имеют переводческие неудачи, связанные с передачей терминов и элементов метаязыка документации. В прагматическом, стилистическом и синтаксическом аспектах переводческие неудачи ведут к ослаблению модальной семантики и директивности, изменению эмоционального и жанрового регистра, а также снижению синтаксической прозрачности и логической связности текста документации по веб-разработке. В результате перевод в меньшей степени обеспечивает освоение технологии как системы взаимосвязанных знаний.

6) Сопоставительный анализ текстов официальной документации к технологиям React JS, Vue.js и MDN Web Docs показывает, что наибольшая концентрация переводческих неудач наблюдается в документации React JS. Это связано с менее последовательным характером принимаемых переводческих решений, несистемностью в передаче коммуникативных параметров оригинала, ослаблением и нивелированием модальности и инструктивности, а также с асинхронностью обновления англоязычной и русскоязычной версий документации. В переводах MDN Web Docs и Vue.js переводческие неудачи также фиксируются, однако носят менее системный характер и реже образуют повторяющиеся серии.

При изучении переводческих неудач, наблюдаемых в переводе англоязычной документации по веб-разработке на русский язык, в качестве **теоретико-методологической базы** используются научные труды отечественных и зарубежных исследователей, среди которых можно выделить следующие значимые для представляемого исследования направления:

– трансфер знаний, его роль в сфере информационных технологий и в переводе: Л. М. Алексеева, С. Л. Мишланова (Алексеева, Мишланова, 2022), А. Б. Бедный (Бедный, 2010), В. З. Демьянков (Демьянков, 2016), И. А. Пильщиков

(Пильщиков, 2016), М. Б. Флек, Е. А. Угнич (Флек, Угнич, 2023), Е. А. Шутаева (Шутаева, 2015) и др.;

– исследования технического текста и технической документации: К. В. Авцынова (Авцынова, 2023), В. Н. Базылев (Базылев, 2010), А. Ю. Горбунов, Л. А. Долбунова (Горбунов, Долбунова, 2015), С. С. Калинина (Калинина, 2017), Л. А. Коняева (Коняева, 2015), Д. В. Семёнов (Семёнов, 2023), Л. Г. Федюченко (Федюченко, 2019) и др.;

– научно-технический перевод и изучение терминологии: В. Н. Базылев (Базылев, 2010), А. В. Жидков (Жидков, 2014), Л. А. Коняева (Коняева, 2015), Ю. С. Котова (Котова, 2013), Е. Д. Полетаева, О. О. Сосновская (Полетаева, Сосновская, 2014), А. Л. Пумпянский (Пумпянский, 1965), Ш. М. Тиллаева (Тиллаева, 2019) и др.;

– интегративный подход к переводу и оценка качества перевода: А. М. Поликарпов (Поликарпов, 2017, 2021, 2025, 2026), В. В. Сдобников (Сдобников, 2015, 2016, 2018), Н. В. Соколова (Соколова, 2020), А. Ф. Шаповалова (Шаповалова, 2019), А. Д. Швейцер (Швейцер, 1988) и др.;

– эрратология и теория переводческих неудач: Л. С. Бархударов (Бархударов, 1975), Ю. Е. Выволокина (Выволокина, 2020), В. Н. Комиссаров (Комиссаров, 1990), Л. К. Латышев (Латышев, 2005), Н. А. Матвеев, Г. А. Шушарина (Матвеев, Шушарина, 2015), Т. М. Софронова, Т. П. Бабак (Софронова, Бабак, 2022), А. А. Щелокова (Щелокова, 2020) и др.;

– исследования современных библиотек и фреймворков веб-разработки, а также характеристик официальной документации по веб-технологиям: С. А. Антонов, А. А. Вуколов, К. А. Кононыхина (Антонов, Вуколов, Кононыхина, 2024), А. А. Байдыбеков, Р. Г. Гильванов, И. А. Молодкин (Байдыбеков, Гильванов, Молодкин, 2020), С. О. Бондаренко (Бондаренко, 2018), В. С. Матвеев (Матвеев, 2020), Е. В. Яровая (Яровая, 2022) и др.

Научная новизна исследования заключается в том, что в нём впервые осуществляется комплексное изучение переводческих неудач, наблюдаемых в

переводе англоязычной документации по веб-разработке на русский язык на основе интегративного подхода к переводу. Новым представляется рассмотрение документации по веб-разработке как разновидности текста, сочетающего черты технического и учебного текста, в аспекте перевода. Впервые переводческие неудачи в текстах перевода данной разновидности систематизируются в лексико-семантическом, прагматическом, стилистическом и синтаксическом ракурсах рассмотрения. Они описываются в фокусе оценки качества перевода с учётом функциональной организации документации и её роли в трансфере специальных знаний.

Теоретическая значимость исследования состоит в уточнении представлений о переводе документации по веб-разработке как о процессе, связанном с передачей и структурированием специального знания в профессиональной цифровой среде. Полученные результаты позволяют конкретизировать возможности изучения категории оценки качества перевода с позиции обнаружения переводческой неудачи применительно к текстам по веб-разработке. Работа вносит вклад в дальнейшую разработку вопросов интегративного переводоведения, оценки качества специального перевода и изучения перевода как одного из механизмов трансфера знаний в сфере информационных технологий.

Практическая значимость исследования определяется возможностью использования его результатов в преподавании дисциплин «Теория перевода», «Практический курс перевода», «Научно-технический перевод», «Перевод в сфере информационных технологий», «Информационные технологии и перевод», «Интегративное переводоведение», а также в курсах, связанных с анализом специального текста и профессиональной коммуникацией. Материалы исследования могут быть использованы при подготовке переводчиков, работающих с англоязычной ИТ-документацией, при разработке рекомендаций по переводу документации в области веб-разработки на русский язык, а также при дальнейшем изучении качества перевода специальных текстов в цифровой среде. Кроме того, результаты работы

могут быть полезны при написании выпускных квалификационных работ и других исследований, посвящённых переводу технической документации и англоязычных текстов в сфере веб-разработки.

Научные положения диссертации **соответствуют паспорту специальности** «5.9.8 – Теоретическая, прикладная и сравнительно-сопоставительная лингвистика», в частности следующим пунктам: 1) лингвистическое переводоведение и его основные направления, языковые и экстралингвистические аспекты перевода; 2) лингвистическое терминоведение.

Степень достоверности результатов исследования определяется системным характером работы, опорой на релевантные отечественные и зарубежные научные труды, теоретико-методологической обоснованностью положений, выносимых на защиту, а также использованием корпуса англоязычных оригиналов и русскоязычных переводов официальной документации по веб-разработке, включающего англоязычные тексты документации к React JS, Vue.js и MDN Web Docs и их переводы на русский язык. Достоверность обеспечивается последовательным применением комплекса методов, соответствующих цели и задачам исследования, непротиворечивостью полученных результатов и их соотнесённостью с общей логикой построения текста диссертационного исследования.

Личный вклад диссертанта диссертанта заключается в самостоятельном формировании теоретической базы исследования, отборе и анализе англоязычных оригиналов и русскоязычных переводов официальной документации по веб-разработке, выявлении и систематизации переводческих неудач, а также в описании их в лексико-семантическом, прагматическом, стилистическом и синтаксическом аспектах. Диссертантом также самостоятельно определены принципы анализа исследуемого материала и обобщены результаты, связанные с оценкой качества перевода документации по веб-разработке в фокусе трансфера специального знания.

Апробация результатов исследования. Основные положения и результаты диссертационного исследования представлены в 8 научных работах,

зарегистрированных в базе данных РИНЦ, в том числе в 3 статьях, опубликованных в журналах, рекомендованных ВАК при Министерстве науки и высшего образования РФ.

Основные положения и результаты диссертации были представлены на международных, всероссийских и региональных научных, научно-практических конференциях и форумах: I Всероссийская научно-практическая конференция «Актуальные вопросы цифровой лингвистики и лингводидактики» (Ставрополь, Северо-Кавказский федеральный университет (СКФУ), 2023), XIII Всероссийская научно-практическая конференция молодых учёных «Язык. Культура. Литература» в рамках Ломоносовских чтений молодых учёных Архангельской области (Архангельск, Северный (Арктический) федеральный университет им. М.В. Ломоносова (САФУ), 2024), XII Всероссийская научно-практическая конференция «Развитие Севера и Арктики: формирование и сохранение традиционных российских духовно-нравственных ценностей» (Архангельск, САФУ, 2024), I Международный лингвоэкологический форум «Вызовы современности и тенденции развития полимодальной коммуникации» (Архангельск, САФУ; Бухара, Бухарский государственный университет, 2024), XIII Всероссийская научно-практическая конференция «Развитие Севера и Арктики: историческая память и социально-гуманитарные вызовы времени» (Архангельск, САФУ, 2025), XIV Всероссийская научно-практическая конференция молодых учёных «Язык. Культура. Литература» в рамках Ломоносовских чтений молодых учёных Архангельской области (Архангельск, САФУ, 2025), Международная научно-практическая конференция «Иностранный язык в сфере профессиональной коммуникации в условиях реальной и виртуальной среды» (Екатеринбург, Уральский федеральный университет им. первого Президента России Б.Н. Ельцина, 2025), Международный форум «Русистика и тюркология: литературные традиции, культурный трансфер и перевод» (Бухара, Бухарский государственный университет, 2025), II Всероссийская научно-практическая конференция «Цифровые практики в лингвистике и образовании»

(Ставрополь, СКФУ, 2025), XV Всероссийская научно-практическая конференция молодых учёных «Язык. Культура. Литература» в рамках Ломоносовских чтений молодых учёных Архангельской области (Архангельск, САФУ, 2026).

Работа также прошла апробацию на публичном представлении диссертационных исследований по прикладной лингвистике аспирантов и соискателей федеральных университетов РФ в рамках деятельности Ассоциации «Консорциум федеральных университетов по научно-образовательному взаимодействию в сфере прикладной лингвистики и профессиональной педагогики „Цифровые практики в лингвистике и образовании”» 20 мая 2026 года.

Объём и структура работы. Диссертация состоит из введения, трёх глав, заключения, списка использованной научной литературы (182 наименование, из которых 69 на иностранных языках), списка словарей и справочников, списка источников эмпирического материала. Общий объём работы составляет 231 страница.

Во введении обосновываются выбор темы исследования, её актуальность и научная новизна, определяется степень разработанности проблемы, формулируются объект, предмет, цель и задачи работы, представляются методы исследования, его теоретико-методологическая база, материал исследования, гипотеза, положения, выносимые на защиту, а также раскрываются теоретическая и практическая значимость диссертации, степень достоверности её результатов, личный вклад диссертанта и сведения об апробации работы.

В первой главе рассматриваются теоретические основания исследования. В ней анализируются подходы к изучению трансфера знаний в сфере информационных технологий, определяется роль веб-разработки в данном процессе, характеризуются технические тексты и документация по веб-разработке как особый тип специального текста, рассматриваются особенности перевода документации данного типа, а также уточняются понятия переводческой ошибки и переводческой неудачи. Особое внимание уделяется методологическим возможностям интегративного

переводоведения и обоснованию использования категории переводческой неудачи применительно к переводу англоязычной документации по веб-разработке на русский язык.

Во второй главе представлена характеристика материала исследования. В ней рассматриваются документации React JS, Vue.js и MDN Web Docs как ключевые источники современного знания в области веб-разработки, анализируются история и значимость соответствующих технологий, структура их официальной документации, особенности англоязычных оригиналов и русскоязычных переводов, а также выявляются факторы, влияющие на качество перевода и доступность документации для русскоязычного реципиента.

В третьей главе осуществляется практический анализ переводческих неудач, выявленных в англоязычной документации по веб-разработке и её переводах на русский язык. В главе описываются материал и методика исследования, рассматриваются лексико-семантические, прагматические, стилистические и синтаксические переводческие неудачи, выявляются особенности их проявления в документации React JS, Vue.js и MDN Web Docs, а также определяется их влияние на функциональную состоятельность перевода как средства трансфера специальных знаний. Каждая из глав сопровождается выводами.

Работа завершается заключением, в котором обобщаются результаты исследования.

1. Трансфер знаний в ИТ-сфере и техническая документация как важный его элемент

1.1. Трансфер знаний и его роль в развитии информационных технологий

Трансфер знаний – процесс, обеспечивающий передачу и сохранение интеллектуальных ресурсов, что в условиях ускоряющегося научно-технического прогресса и интернационализации знания играет важнейшую роль. Понятие трансфера как такового появилось ещё в античные времена (от греч. μεταφορά). Под трансфером понималось «постоянное изменение», «движение» (Chambers, 1983, с. 789). В последние десятилетия понятие трансфера знаний стало важным компонентом академических и прикладных исследований, поскольку оно связано не столько с передачей информации, сколько с комплексной передачей опыта, навыков, культурных и профессиональных традиций. Как отмечает Е.А. Шутаева, инновационная государственная политика XXI в. характеризуется всё возрастающим запросом осуществлять приумножение и распространение новых знаний, а также стимулировать процессы по созданию и внедрению основанных на этих знаниях технологий (Шутаева, 2015, с. 2). В сфере профессиональной коммуникации трансфер знаний рассматривается как механизм, позволяющий интегрировать накопленные данные и навыки из различных дисциплин, а также адаптировать их к новым задачам, что особенно важно в условиях цифровой трансформации и изменения рыночных условий. Обосновывается такой интерес к трансферу знаний «невозможностью обеспечить конкурентное преимущество производимой

продукции, [в том числе нематериальной], в условиях рыночной экономики» (Заварзин, 2013, с. 1).

Теоретические подходы к трансферу знаний отражают эволюцию взгляда на то, как знания могут создаваться, трансформироваться, сохраняться и передаваться. В коллективной монографии, посвящённой трансферу знаний, Л. М. Алексеева и С. Л. Мишланова указывают на то, что он имеет многосоставную структуру и включает аспекты коммуникации, управления и практического применения (Алексеева, Мишланова, 2022, с. 14). Авторы отмечают, что понятие трансфера знаний обогащается благодаря его многозначной роли в когнитивной науке и инновационных исследованиях (там же, с. 9). Из этого можно сделать вывод, что трансфер знаний связан помимо прочего с созданием условий для формирования новой, интегративной базы знаний, способной решать комплексные задачи в условиях меняющейся научной и социальной реальности.

Продолжая эту мысль, можно отметить, что процесс трансфера знаний выходит за пределы передачи информации и приобретает характер динамического обмена, способствующего развитию когнитивной среды. Ряд отечественных и зарубежных исследований указывают на тот факт, что формирование новой, интегративной базы знаний требует эффективных механизмов передачи и активного участия субъектов, готовых адаптировать полученную информацию в контексте актуальных вызовов (Васюченок, 2015; Демьянков, 2016; Argote, 2000; Ward, 2009; Bacon, 2021).

В этом смысле важную роль играют цифровые технологии, обеспечивающие доступ к большим массивам данных, а также способы их структурирования и интерпретации (Гительман, 2020). Взаимодействие между искусственным интеллектом, системами управления знаниями и экспертным сообществом создаёт условия для непрерывного обновления и совершенствования имеющихся знаний, расширяя горизонты научного поиска (Пильщиков, 2016, с. 210).

Кроме того, трансфер знаний приобретает всё большую значимость в условиях междисциплинарности (Adler et al., 2018, p. 181). Совмещение различных методов и

концепций способствует разработке инновационных решений, позволяя адаптировать знания к быстро изменяющимся технологическим и социальным реалиям (Терехина, 2023, с. 160; Божева, 2018, с. 38).

Зарубежные исследователи также подчеркивают значимость трансфера знаний как фактора развития научных и технологических инноваций. В частности, исследователи отмечают роль международного обмена знаниями в формировании глобальных стратегий и политических решений (Argote, 2024, p. 5), а также влияние коллективных механизмов трансфера технологий на взаимодействие науки и бизнеса (Tien, 2023, p. 8; Massaro, 2019, p. 269). Этот процесс особенно актуален для управления знаниями в организации, где гибкость и доступность информации напрямую влияют на эффективность работы.

Как было сказано выше, на первых этапах становления термин «трансфер» ассоциировался с движением и перемещением материальных или нематериальных объектов из одной сферы в другую. Однако в рамках современных научных концепций трансфер знаний трактуется гораздо шире: как процесс, включающий когнитивное взаимодействие, передачу опыта, навыков и специализированной информации (Заварзин, 2013, с. 1). Примеры таких процессов можно встретить в междисциплинарных и трансдисциплинарных исследованиях, когда знание передается от одной области знания к другой (Bacon, 2023). В отличие от линейного понимания, присущего начальным этапам научной мысли о трансфере, современные подходы описывают трансфер знаний как сложный процесс, включающий адаптацию, трансформацию и создание нового знания на основе уже имеющегося (Алексеева, Мишланова, 2022, с. 15; Ferrer-Serrano, 2026, p. 5238).

Еще один подход к пониманию трансфера знаний разработан М.Б. Флеком и Е.А. Угнич. Они говорят про сетевой подход, который акцентирует внимание на взаимодействии различных субъектов – организаций, университетов, исследовательских центров и экспертных сообществ. В рамках этого подхода трансфер знаний рассматривается как процесс, происходящий в сложных сетях, где

важную роль играет доверие между участниками и их способность к восприятию и адаптации новых знаний (Флек, Угнич, 2023, с. 89).

Другой значимый подход – клинический, основанный на концепции практического обучения. В этом случае знания передаются через непосредственное взаимодействие с реальными кейсами и проблемами, что особенно актуально для университетов и исследовательских центров. Такой метод позволяет студентам и специалистам применять знания в реальных условиях, что способствует эффективному трансферу знаний (Базжина, 2024, с. 88).

А. Б. Бедный в своей монографии «Основы организации университетского трансфера знаний» рассматривает трансфер знаний с точки зрения институционального подхода, в котором трансфер знаний представляет собой процесс, регулируемый институциональными механизмами, такими как государственная политика, корпоративные стратегии и образовательные программы. В этом контексте анализируются факторы, влияющие на успешность передачи знаний, включая организационную культуру, нормативные барьеры и экономические условия (Бедный, 2010, с. 15).

Исследователи трансфера знаний выделяют следующие функции трансфера знаний:

Во-первых, трансфер знаний способствует сохранению и накоплению знаний, обеспечивая их передачу от одного поколения специалистов к другому. Это особенно важно в академической среде, где исследования и открытия требуют преемственности и систематизации накопленного опыта (Алексеева, Мишланова, 2022, с. 68; Бедный, 2010, с. 24).

Во-вторых, трансфер знаний связан с адаптацией и трансформацией информации. В процессе передачи знания модифицируются, дополняются новыми данными и адаптируются к изменяющимся условиям. Это особенно актуально в условиях стремительного развития технологий, когда устаревшие методы заменяются инновационными подходами (Аманов, 2022, с. 24; Marchiori, 2020).

Более того, как отмечают, Л.М. Алексеева и С.Л. Мишланова, процесс трансфера знаний дает возможность интегрировать новые знания в уже существующие структуры, делая их более гибкими и способными реагировать на вызовы современности (Алексеева, Мишланова, 2022, с. 68).

Третья важная функция трансфера знаний – стимулирование инноваций. Передача знаний между различными дисциплинами и секторами экономики способствует созданию новых идей, технологий и продуктов. Взаимодействие между научными учреждениями, бизнесом и государственными структурами позволяет ускорить процесс внедрения инноваций и повысить их эффективность (Суховаров, 2016; Lundmark, 2023).

Кроме того, трансфер знаний выполняет образовательную функцию, обеспечивая подготовку специалистов, способных работать в условиях быстро меняющегося мира. Университетские программы, корпоративное обучение и профессиональная переподготовка являются важными элементами этого процесса, позволяя специалистам осваивать новые навыки и адаптироваться к современным требованиям рынка (Бедный, 2010).

Л.М. Алексеева и С.Л. Мишланова также подчеркивают, что трансфер знаний включает в себя такие механизмы, как моделирование, «коллаборативный перевод», а также развитие транссемиотических компетенций, необходимых для эффективного восприятия и адаптации знаний в новых контекстах (Алексеева, Мишланова, 2022, с. 80). Моделирование позволяет создавать схематические представления знания, упрощающие его передачу и восприятие. «Коллаборативный перевод» способствует адаптации знаний в межкультурной и междисциплинарной среде, что особенно актуально в условиях глобализации и цифровой трансформации (Алексеева, Мишланова, 2022, с. 80). Зарубежные исследователи трансфера знаний во многом согласны с таким подходом (Ward, 2009; Herfeld, 2019; Domans, 2021).

В последние десятилетия трансфер знаний стал одним из основополагающих процессов в сфере информационных технологий (ИТ) (Шутаева, 2015). Это

обусловлено тем, что сложные программные системы, постоянно обновляющиеся технологические платформы и необходимость быстрого внедрения инноваций требуют высокого уровня гибкости и приспособляемости знаний к новым задачам (Argote, 2000, p. 6]. Понятным в связи с этим является утверждение о том, что трансфер знаний в сфере информационных технологий охватывает как технические, так и управленческие аспекты: от обмена новыми методами кодирования до внедрения стандартов информационной безопасности (там же).

Сфера ИТ обладает рядом уникальных черт, которые определяют особенности осуществляемого в ней трансфера знаний. Во-первых, сфера информационных технологий характеризуется высокой скоростью изменений: новые технологии, такие как искусственный интеллект и машинное обучение, постоянно модернизируются, и специалистам необходимо осваивать их, даже еще до того, как они станут частью повседневной практики (Флек, Угнич, 2023; Storey, 2025; Xin, 2024). Во-вторых, технологии требуют от специалистов способности адаптировать существующие знания под конкретные задачи и контексты (Фёдоров, 2008, с. 49). В результате этого трансфер знаний в ИТ становится сложным процессом, в котором участвуют множество специалистов, работающих над проектами в различных странах и организациях.

Примером успешного трансфера знаний в ИТ является, на наш взгляд, распространение методов разработки программного обеспечения, таких как Agile и DevOps, относящихся к методологии взаимодействия разработчиков, тестировщиков и других ИТ-специалистов в команде. Эти подходы стали международными стандартами в ИТ-отрасли, помогая улучшить управление проектами, повысить производительность и обеспечить гибкость (Dybå, 2008; Aouni, 2025). Однако, на наш взгляд, внедрение этих подходов требует значительных усилий по обучению специалистов, что возможно только через организованный трансфер знаний: обмен опытом, обучение и адаптацию данных подходов под специфику каждой конкретной команды. На это указывают и исследования в данной области, например, в работе

А.С. Ермакова рассматриваются этапы развития методологии DevOps и её влияние на процессы управления проектами. Автор подчёркивает, что успешное внедрение DevOps требует активного трансфера знаний между командами, что способствует повышению эффективности работы и снижению рисков (Ермаков, 2020, с. 91).

В научной статье Т.С. Забело и Е.И. Полоско анализируются методологии разработки программного обеспечения. Отмечается, что эти подходы обеспечивают гибкость и адаптивность команд, а их успешное внедрение возможно только при наличии эффективных механизмов передачи знаний и постоянного обучения специалистов (Забело, Полоско, 2024, с. 251).

Также значительную роль в трансфере знаний в области ИТ играют сообщества разработчиков, такие как GitHub и Stack Overflow. Эти платформы позволяют специалистам по всему миру делиться кодом, обсуждать проблемы и предлагать решения, создавая уникальную базу знаний, доступную всем заинтересованным (Temizkan, 2024; Blanco, 2020). Отметим также, что такой трансфер знаний выходит за рамки чисто технической информации: он включает обучение культурным и профессиональным особенностям работы в международных командах, что делает процесс обмена знаниями более комплексным.

Важно отметить, что трансфер знаний в ИТ-секторе также включает внедрение знаний, связанных с вопросами безопасности данных и защиты конфиденциальности (Белашова, 2017). Действительно, поскольку современные компании часто работают с большими объёмами данных, необходимость обмена лучшими практиками в области кибербезопасности становится неотъемлемой частью работы ИТ-команд. Трансфер знаний в этой области помогает быстро адаптировать новые методы защиты данных к конкретным нуждам организации и своевременно реагировать на возникающие угрозы.

Несмотря на значимость трансфера знаний в ИТ, он сталкивается с рядом вызовов и ограничений. Во-первых, барьеры могут возникать из-за недостатка единого стандарта или структуры для передачи знаний (Сазонова, 2025; Фёдоров,

2008; Флек, Угнич, 2023). Например, при внедрении новой технологии специалисты из разных отделов компании могут интерпретировать ее особенности по-своему, что может привести к несоответствию в практиках и ошибкам в реализации (Фёдоров, 2008).

Кроме того, для успешного трансфера знаний требуются время и ресурсы, которые не всегда есть в наличии (Nidhra, 2013). В условиях высокой конкуренции и постоянных обновлений, характерных для ИТ-сферы, организациям, занимающимся разработкой веб-страниц, например, часто не хватает ресурсов на формальные процессы обмена знаниями (Калабина, Беляк, 2021). Многие компании вынуждены полагаться на неформальные каналы передачи знаний, такие как обсуждения в рабочих чатах или встречах по проектам. Это может ограничивать доступ к важной информации и создавать проблемы для новых сотрудников, которые не могут сразу включиться в работу без должного уровня подготовки (Узюмова, 2019).

Также немаловажные ограничения связаны с культурными различиями, которые могут влиять на восприятие и адаптацию знаний (Zaidman, 2009). В международных IT-компаниях специалисты из разных стран могут иметь разные подходы к решению задач и управления проектами, что создает дополнительные сложности при обмене знаниями. Например, одна и та же технология может восприниматься по-разному в зависимости от особенностей культуры отдельных представителей команды разработчиков, что требует дополнительных усилий по адаптации знаний к контексту, в котором они будут применяться.

Подводя промежуточный итог предшествующему обсуждению, можно сказать, что трансфер знаний является важнейшим процессом, который обеспечивает передачу продуктов интеллектуальной деятельности в условиях глобализации и цифровой трансформации.

1.2. Роль веб-разработки как области информационных технологий в трансфере знаний

Трансфер знаний в сфере веб-разработки обладает рядом характеристик, которые определяются спецификой данной области, а также быстрыми темпами её развития. С учётом комплексного характера веб-разработки, которая объединяет такие направления, как программирование, проектирование интерфейсов, работа с базами данных и обеспечение безопасности, важно адаптировать их к меняющимся условиям и требованиям конкретных проектов. Современные исследователи отмечают, что трансфер знаний включает как обмен технологиями, так и трансляцию «личного понимания и опыта» между специалистами, что делает его важным элементом командной работы и проектного взаимодействия (Алексеева, Мишланова, 2022).

В статье Н.И. Газизуллина и И.Е. Плещинской рассматривается влияние прогрессивного веб-приложения (Progressive Web Apps – PWA) на веб-разработку и удобство работы пользователей с сайтами. PWA – особый тип веб-приложений, которые могут работать без интернета, отправлять уведомления и загружаться быстрее обычных сайтов. Авторы подчёркивают, что успешное внедрение таких технологий зависит от эффективного обмена знаниями между разработчиками, что подтверждает важность передачи опыта в этой сфере (Газизуллин, Плещинская, 2020, с. 622].

Особенность трансфера знаний в сфере веб-разработки заключается прежде всего в том, что многие проекты требуют быстрой интеграции новых сотрудников в существующие команды и рабочие процессы и совместной работы специалистов с различной подготовкой. Эффективный трансфер знаний позволяет минимизировать временные затраты на обучение, ускорить процесс вхождения в проект и повысить

продуктивность всей команды (Матвеева, 2019]. Одной из важных задач в таких случаях становится структурированная передача информации и систематизация знаний, которая позволяет облегчить процесс обучения и быстрее включить сотрудников в тот или иной проект (там же).

Необходимо подчеркнуть, что в веб-разработке широко используются методы управления знаниями, такие как документирование процессов, использование репозиториев для хранения и контроля версий кода (Fregnan, 2023; Zolkifli, 2018), а также регулярные обсуждения в формате код-ревью (т.е. командной работы по улучшению структуры и читаемости кода) (Арсеньев, 2022). Эти инструменты служат для создания общего информационного пространства, в котором каждый член команды может ориентироваться на существующие стандарты и подходы.

Документация, создаваемая для веб-разработчиков, участвует в трансфере знаний, особенно в условиях быстрого развития инструментов и фреймворков (каркасов для написания веб-приложений, определяющих структуру, задающих правила и предоставляющих необходимый набор инструментов для разработки) (Прияцелюк, 2018)) таких как React (React: The Library for Web and Native User Interfaces) и Angular (Angular: The Web Development Framework for Building Modern Apps). Этот тип документации представляет собой официальные руководства, учебные курсы и мануалы, размещаемые на сайтах, посвящённых данным технологиям, и предоставляет пользователям структурированную информацию, помогающую быстро освоить базовые принципы работы с новой технологией. Подобная документация не привязана к конкретным проектам, но создается с целью обучить широкую аудиторию основным методам и практикам использования технологии, облегчая процесс её освоения (MDN Web Docs).

Роль документации по веб-разработке в трансфере знаний особенно велика по нескольким причинам. Во-первых, она создаётся разработчиками технологии, которые обладают наибольшей глубиной понимания и могут предложить наиболее точные и актуальные сведения о её функциональности. Например, разработчики

React предлагают учебные материалы, демонстрирующие, как лучше всего использовать эту библиотеку в различных ситуациях (React: The Library for Web and Native User Interfaces). Такие учебные курсы часто содержат подробные объяснения, пошаговые инструкции и примеры кода, которые помогают веб-разработчикам понять логику и возможности новой технологии, а также её совместимость с другими популярными инструментами (Meng, 2019).

Во-вторых, документация, создаваемая для веб-разработчиков, обычно содержит концептуальные основы, описывающие принципы проектирования и рекомендации по использованию технологий. Это помогает разработчикам понимать глубинные идеи, лежащие в основе инструментов и методологий, что способствует более эффективному трансферу знаний и профессиональному развитию. В этом смысле документация по веб-разработке призвана воздействовать на потребителей как средство для передачи знаний о подходах и парадигмах, заложенных в конкретной технологии (там же).

Официальная документация по веб-разработке также выполняет роль стандарта, на который ориентируются разработчики и команды по всему миру (там же). Доступ к единому источнику информации снижает вероятность ошибок и интерпретационных различий, которые могли бы возникнуть, если бы разработчики пользовались различными неофициальными источниками. Унифицированная документация создает основу для формирования общих подходов и лучших практик, позволяя разработчикам, работающим в разных организациях и странах, вырабатывать схожие методы работы с той или иной технологией в области проектирования интерфейсов, организации архитектуры веб-приложений, написания и оптимизации кода, а также взаимодействия с базами данных и серверными компонентами (Kotstein, 2023).

Кроме того, такая документация часто обновляется и дополняется с выпуском новых версий фреймворков и библиотек. Фреймворки и библиотеки, хотя и схожи по назначению, имеют принципиальные различия. Библиотека предоставляет набор

готовых функций, которые разработчик может использовать по своему усмотрению, вызывая их в нужный момент. Фреймворк же задаёт общую архитектуру приложения и управляет его потоком, предоставляя разработчику определённые точки для внесения изменений (Антонов, 2024).

Благодаря этому разработчики могут следить за развитием технологии и быстро адаптироваться к изменениям, что особенно важно в контексте веб-разработки, где инструменты и подходы меняются быстро. Официальные учебные материалы и руководства, дополненные новыми функциями и методами, служат механизмом для распространения этих обновлений, поддерживая актуальность знаний и обеспечивая их соответствие последним стандартам.

Наконец, официальная документация также создаёт условия для самообучения, что позволяет разработчикам освоить технологию в удобном для них темпе. Она обеспечивает доступ к структурированным материалам для специалистов разного уровня подготовки – от новичков до продвинутых пользователей, предлагая специализированные разделы и курсы для каждой категории. Это делает процесс трансфера знаний более гибким и доступным, позволяя веб-разработчикам с различным уровнем опыта находить необходимые им материалы и таким образом совершенствовать свои умения и навыки.

Особое значение в трансфере знаний в сфере веб-разработки имеют процессы упомянутого выше код-ревью и менторства (Fregnan, 2023). Код-ревью является практическим средством передачи знаний и умений, при котором более опытные разработчики проверяют и комментируют код своих коллег, помогая им улучшать качество принимаемых решений (Данилина, 2024; Fregnan 2023). Такой подход позволяет новичкам быстрее понять архитектуру системы, а также принципы и стандарты кодирования. Это соответствует представленной в монографии Л.М. Алексеевой и С.Л. Мишлановой идее о том, что трансфер знаний становится действительно эффективным, когда он включает передачу опыта, связанного с внутренней логикой и культурой организации (Алексеева, Мишланова, 2022).

Код-ревью позволяет реализовать на практике концепцию «интерактивного обучения», в котором каждая проверка кода становится способом повысить уровень компетентности команды и адаптировать новые идеи под конкретные задачи проекта (Fregnan, 2023). Более того, в процессе код-ревью специалисты учатся мыслить критически и анализировать различные аспекты разработки, что способствует глубокому пониманию используемых технологий и позволяет находить оптимальные решения для решаемых задач (там же).

Менторство в веб-разработке – это процесс наставничества, при котором опытные разработчики помогают менее опытным специалистам освоить технологии, методологии и лучшие практики. Оно играет важную роль в трансфере знаний, обеспечивая передачу стратегического мышления, необходимого для эффективной работы в команде. Персонализированное обучение в рамках менторства позволяет адаптировать процесс передачи знаний под уровень и потребности ученика, помогая ему быстрее осваивать сложные концепции (Цибизова, 2017).

Веб-разработка представляет собой динамично развивающуюся область ИТ-технологий, в которой значительную роль играют открытые сообщества и форумы для обмена знаниями, такие как Stack Overflow, Dev.to и Reddit. Данные платформы позволяют разработчикам задавать вопросы, делиться кодом и решениями, а также находить ответы на возникающие проблемы (Fregnan, 2023).

Как отмечают Л.М. Алексеева и С.Л. Мишланова, успешный трансфер знаний предполагает «создание благоприятной среды для обмена опытом» и использования новых подходов, проверенных другими специалистами в разных условиях (Алексеева, Мишланова, 2022, с. 25). Разработчики, работающие над сложными задачами, могут быстро находить информацию, проверять различные методы и адаптировать найденные решения к конкретным требованиям проекта. Глобальные сообщества веб-разработчиков играют важную роль в трансфере знаний, облегчая доступ к накопленным знаниям и помогая специалистам профессионально развиваться.

Совместное обучение и обмен опытом также происходят посредством проведения хакатонов (своего рода состязаний, во время которых за определенный срок решаются поставленные перед ними задачи) и мастер-классов, на которых специалисты в области веб-разработки могут знакомиться с новыми технологиями и обмениваться практическими решениями в условиях, приближенных к реальной работе. В хакатонах часто участвуют представители различных областей веб-разработки, что позволяет создать междисциплинарное взаимодействие, при котором каждый может внести свой вклад в решение задач (Васильева, 2024). Такие формы коллективного обучения способствуют «формированию трансдисциплинарных компетенций» и позволяют участникам обмениваться знаниями в более гибких и неформальных условиях (Алексеева, Мишланова, 2022, 18).

Одним из наиболее сложных аспектов трансфера знаний является адаптация технологий и методов под конкретные требования компании или проекта. Каждый проект в веб-разработке может иметь свои уникальные требования, будь то использование специфических языков программирования, подходов к дизайну и доступности или требований безопасности.

Л.М. Алексеева и С.Л. Мишланова отмечают, что трансфер знаний должен учитывать возможность развития компетенций и адаптации знаний к новым условиям (Алексеева, Мишланова, 2022, с. 67), что особенно важно при работе с веб-проектами, поскольку каждый проект требует гибкости и индивидуального подхода Адаптация знаний проявляется также в необходимости учитывать культурные и языковые особенности, если веб-разработка ведётся в международной команде. Стандарты и подходы, принятые в одной стране или организации, могут отличаться от тех, которые применяются в других регионах мира или организациях. Это требует от разработчиков умения адаптировать знания и учитывать особенности разных культур и рынков, что позволяет повысить качество продукта и создать более гибкую структуру работы.

Таким образом, трансфер знаний в веб-разработке представляет собой многоуровневый процесс, включающий технические и социальные аспекты, которые обеспечивают эффективное взаимодействие между специалистами, поддержание высокого уровня качества разработки и адаптацию технологий к конкретным требованиям. Веб-разработчики используют различные методы, от документирования и код-ревью до совместных мастер-классов и глобальных сообществ, что помогает создать прочную основу для коллективного обучения и обмена знаниями.

1.3. Роль трансфера знаний в развитии индустрии перевода

Трансфер знаний является фундаментальным элементом современного перевода, определяя его успешность в адаптации информации к культурным и профессиональным контекстам. Перевод уже давно вышел за пределы подбора эквивалентов между языковыми единицами, превращаясь в сложный когнитивный и коммуникативный процесс. Этот процесс требует глубокого понимания контекста, культуры и междисциплинарного подхода, а также способности интегрировать знания из различных сфер (Оберюхтина, 2022, с. 151).

Исследования в области лингвистики подтверждают, что эффективный перевод невозможен без трансфера знаний, который охватывает как языковые, так и экстралингвистические аспекты (Золян, 2019). Учитывая многоуровневость переводческой деятельности, современный переводчик должен владеть методами анализа и синтеза информации, обеспечивая точность и адекватность смысловой передачи (Бочавер, Фещенко, 2017). Перевод становится полноценной

коммуникацией, способной учитывать сложные когнитивные механизмы и социальные факторы.

Современные исследования в области переводоведения подчеркивают, что перевод – это, помимо прочего, механизм осмысления и преобразования знаний (Алексеева, Мишланова, 2022; Голубева, 2022; Люсый, 2016; Chen Eoyang, 2010). В этом контексте трансфер знаний выполняет несколько важных функций: он способствует адаптации смысла, минимизирует потери при передаче информации и помогает учитывать культурные особенности (Золян 2019). К примеру, перевод специализированных текстов (юридических, медицинских, технических) требует точности и, что более важно, понимания концептуальной структуры дисциплины, ее терминологии и профессиональных стандартов.

Другой важный аспект трансфера знаний в переводе – его полимодальный характер. Современные тексты часто включают визуальные и аудиальные компоненты, и переводчик должен учитывать эти элементы при адаптации материала.

Одним из значимых подходов к изучению трансфера знаний в переводе является интегративное переводоведение, разрабатываемое в Северном (Арктическом) федеральном университете имени М. В. Ломоносова под руководством А. М. Поликарпова. В рамках данного направления переводческая деятельность рассматривается как динамическая целенаправленная система, на которую воздействует совокупность внутренних и внешних факторов (Поликарпов, 2017, с. 7). При этом интегративное переводоведение формируется на пересечении целого ряда исследовательских перспектив и связано с развитием переводоведения под воздействием культурологической, герменевтической, семиотической, деятельностной, когнитивно-дискурсивной и некоторых других интеграционных научных парадигм (там же, с. 8). Такой подход позволяет описывать перевод в более широкой исследовательской рамке и учитывать те смысловые, дискурсивные,

интерпретационные и деятельностные параметры, которые определяют характер переводческого результата.

Существенное место в концепции А. М. Поликарпова занимает мысль о необходимости комплексного изучения перевода с учётом творческой личности переводчика, многослойности его мышления, текстоцентрического принципа и комплексной оценки качества перевода (там же, с. 11). Особое значение в этой связи приобретает предложенный исследователем принцип «многоканальности» мышления переводчика, предполагающий интеграцию сознательных ментальных действий с бессознательными операциями. Речь идёт о таком понимании переводческой деятельности, при котором исследователь стремится выявить, каким образом различные факторы переводческого процесса взаимодействуют друг с другом и как это взаимодействие отражается в итоговом качестве перевода (там же).

А. М. Поликарпов подчёркивает, что становление интегративного переводоведения связано с рядом предпосылок. К числу таких предпосылок он относит признание переводоведения самостоятельной научной отраслью междисциплинарного или трансдисциплинарного характера, формирование новых интеграционных научных парадигм, выдвижение интегрирующего подхода к переводу как особого научного направления, а также попытки моделирования процесса перевода с интегративных, то есть холистических, позиций (там же, с. 8). В этом контексте современная теория перевода действительно выходит за пределы сугубо лингвистического анализа и всё чаще рассматривает перевод как многомерное явление, требующее привлечения данных разных областей знания.

Настоящее исследование опирается на идеи интегративного переводоведения, поскольку оно позволяет рассматривать перевод документации по веб-разработке как процесс трансфера специальных знаний, зависящий от целого комплекса факторов. Такой подход позволяет учитывать свойства самой документации как особой разновидности технических текстов, специфику цифровой профессиональной среды,

особенности восприятия переводного текста адресатом, а также качество переводческого результата.

Коммуникативно-функциональный подход к переводу, разработанный В.В. Сдобниковым, представляет собой еще одну значимую концепцию современного переводоведения. В отличие от традиционных текстоцентрических моделей, этот подход рассматривает перевод как динамический процесс, в котором ключевую роль играет коммуникативная цель текста и его функциональное предназначение (Сдобников, 2015, с. 44). Основная идея коммуникативно-функционального подхода заключается в том, что перевод – это прежде всего сложный акт межъязыковой и межкультурной коммуникации, требующий учета прагматических факторов, намерений автора и ожиданий целевой аудитории.

Коммуникативно-функциональная модель перевода основывается на том, что перевод всегда осуществляется в конкретной коммуникативной ситуации, где пересекаются различные виды деятельности участников процесса. Он выполняется с определенной целью, которая задается инициатором перевода и может не совпадать с интересами конечного получателя. Кроме того, специфика коммуникативной ситуации влияет на стратегию перевода, а коммуникативный эффект оригинала и перевода может различаться. Таким образом, переводчик активно адаптирует информацию, обеспечивая успешное взаимодействие между разноязычными коммуникантами и способствуя эффективному обмену знаниями (там же).

В.В. Сдобников подчеркивает, что переводчик должен ориентироваться на коммуникативную ситуацию, в которой функционирует текст (Сдобников, 2016, с. 45). Одним из ключевых аспектов коммуникативно-функционального подхода является концепция прагматической адаптации, согласно которой переводчик должен обеспечивать восприятие информации целевой аудиторией. Это особенно актуально в условиях глобализации, когда перевод становится инструментом межкультурного взаимодействия.

Коммуникативно-функциональный подход тесно связан с концепцией трансфера знаний, поскольку переводчик адаптирует текст с учетом специфики целевой аудитории. В этом контексте перевод можно рассматривать как процесс интеграции знаний, в котором участвуют и языковые, и экстралингвистические факторы. В.З. Демьянков подчеркивает, что переводчик должен владеть когнитивными техниками трансфера знаний, позволяющими минимизировать потери смысла и обеспечивать точность передачи информации (Демьянков, 2016, с. 30).

В.В. Сдобников рассматривает перевод как процесс, включающий взаимодействие различных знаковых систем. В современных условиях перевод все чаще выходит за рамки работы с чисто текстовыми материалами и охватывает полимодальные тексты, включающие визуальные и аудиальные элементы. Это требует от переводчика транссемиотических компетенций, позволяющих учитывать невербальные компоненты коммуникации и адаптировать их к целевой культуре (Сдобников, 2015, с. 78).

Развивая коммуникативно-функциональную модель перевода, В.В. Сдобников уделяет большое внимание вопросам оценки качества перевода, подчеркивая, что оно должно основываться прежде всего на коммуникативной эффективности. Он предлагает рассматривать перевод как коммуникативно-функциональный аналог оригинала, в котором важную роль играет соответствие переведенного текста ожиданиям целевой аудитории и его способность выполнять ту же функцию, что и оригинал (Сдобников, 2018, с. 71). Это особенно важно в синхронном переводе, где переводчик должен учитывать интонацию, жесты и другие невербальные элементы коммуникации.

Таким образом, коммуникативно-функциональный подход. разрабатываемый В.В. Сдобниковым и его коллегами, представляет собой современную концепцию переводоведения, ориентированную на учет прагматических, когнитивных и культурных факторов. Этот подход позволяет рассматривать перевод как динамический процесс, в котором трансфер знаний играет ключевую роль в

адаптации и интеграции смыслов. В условиях глобализации и цифровизации данный метод становится особенно актуальным, поскольку он обеспечивает высокое качество коммуникации и способствует эффективному межкультурному взаимодействию. Исследования в этой области подчеркивают важность междисциплинарного подхода и применения инновационных методов, которые помогают переводчикам адаптироваться к новым вызовам (Алексеева, Мишланова, 2019).

Один из ведущих исследователей семиотики и теории культуры Ю. М. Лотман разработал концепцию динамики смыслов, которая также играет важную роль в понимании трансфера знаний в переводе. Он утверждает, что перевод – сложный процесс взаимодействия различных знаковых систем, в ходе которого происходит обогащение смысла (Lotman, 2000, с. 13). В этом контексте перевод можно рассматривать как форму культурного творчества, при которой переводчик выступает активным участником процесса смыслообразования.

Ю.М. Лотман подчеркивает, что каждый язык и культура обладают уникальными средствами выражения, которые невозможно полностью передать в другом языковом пространстве без потерь или изменений. Это связано с тем, что языковые знаки несут в себе как лексическое значение, так и культурные, исторические и социальные коннотации, которые могут существенно отличаться в разных традициях. Таким образом, перевод неизбежно включает в себя элементы адаптации, интерпретации и даже реконструкции исходного текста, что делает его процессом передачи и создания новых смыслов (там же).

Особое внимание Ю.М. Лотман уделяет понятию семиосферы – пространства знаковых систем, в котором происходит смыслообразование. В рамках этой концепции перевод можно рассматривать как процесс пересечения границ между различными семиотическими структурами, что приводит к появлению новых интерпретаций и смыслов. Согласно представляемой концепции, переводчик, работая с текстом, переносит информацию, данную в нем, в другую языковую среду,

и адаптирует ее к новой культурной реальности, что делает перевод динамическим и творческим процессом. В этом смысле перевод можно сравнить с диалогом между культурами, в ходе которого «происходит обмен знаниями, ценностями и мировоззренческими установками» (Лотман, 1992, с. 45).

Концепция динамики смыслов особенно актуальна в условиях глобализации, когда перевод становится способом культурного взаимодействия. Действительно, современные переводчики работают с текстами, которые требуют адаптации к специфике целевой аудитории. В таких случаях переводчик становится медиатором, создающим новые культурные продукты, способные отражать богатство и разнообразие мировых культур (Lotman, 2000).

Таким образом, трансфер знаний становится сегодня основополагающим принципом перевода, определяя его эффективность, точность и адаптацию смыслов. В условиях глобализации и цифровизации значение трансфера знаний в переводе в значительной степени возрастает, требуя от переводчиков высокой компетентности, способности к анализу и интерпретации, а также умения работать с различными знаковыми системами. Современное переводоведение подтверждает, что перевод – это сложный когнитивный и коммуникативный процесс, в котором трансфер знаний служит ключевым механизмом смыслообразования и культурного взаимодействия.

1.4. Место веб-разработки в современной ИТ-индустрии

В современном информационном обществе веб-разработка занимает центральное место, представляя собой комплексный процесс создания и поддержки веб-сайтов. Веб-разработка охватывает широкий спектр деятельности, начиная от

создания дизайна и визуального оформления до написания кода, который обеспечивает функциональность и взаимодействие пользователей с веб-ресурсом (Матвеева, 2019). Суть веб-разработки, как описано в работе А.В. Матвеевой, заключается в профессиональном программировании сайтов с использованием HTML-разметки, а также различных технологий и инструментов для обеспечения их работоспособности, удобства использования и технической стабильности (Матвеева, 2019, с. 536). HTML (англ. hypertext markup language – язык гипертекстовой разметки) является одним из основных инструментов веб-разработки, который определяет структуру содержимого веб-страницы. HTML позволяет организовать текст, изображения, видео и другие мультимедийные элементы на веб-странице таким образом, чтобы пользователи могли легко взаимодействовать с ними. Вместе с CSS (англ. cascading style sheets – каскадные таблицы стилей), особым языком программирования, позволяющим управлять стилями на веб-странице, HTML обеспечивает визуальное оформление и структурирование веб-страниц, делая их привлекательными и удобными для просмотра (Tabares, 2021, p. 126; Иванов, 2018, с. 42).

Задачами веб-разработчика являются создание инфраструктуры для реализации оптимального функционала веб-сайта и обеспечение его эффективной работы и удобства использования для пользователей (Куваев, Антимонов, 2020, с. 1270). Основная цель веб-разработки заключается в том, чтобы предоставить пользователям возможность взаимодействовать с веб-ресурсом без проблем и задержек. Для этого разработчики должны грамотно организовать структуру сайта, учитывая потребности и ожидания целевой аудитории. Они также отвечают за интеграцию визуальных компонентов, которые делают сайт привлекательным и функциональным, обеспечивая интерактивные возможности, такие как анимации, формы обратной связи и другие элементы, способствующие активному взаимодействию пользователей с содержанием веб-страниц. Кроме того, разработчики должны учитывать требования к безопасности, чтобы защитить сайт от

потенциальных угроз и обеспечить сохранность данных пользователей (Куваев, Антимонов, 2020, с. 1270).

Веб-разработка важна в современном мире, поскольку веб-сайты стали неотъемлемой частью повседневной жизни людей. Сайты представляют собой важный канал коммуникации между компаниями и их клиентами, позволяя предоставлять информацию о продуктах и услугах, обеспечивать онлайн-продажи, поддерживать контакт с аудиторией через блоги и социальные сети, а также обеспечивать доступ к различным онлайн-ресурсам (Двуреченский, 2023). Другими словами, веб-разработка стала основой цифровой инфраструктуры, обеспечивая функционирование множества сервисов и технологий, используемых как в бизнесе, так и в частной жизни. Благодаря ей создаются платформы для онлайн-образования, государственных услуг, электронных библиотек, финансовых систем и социальных сетей. Взаимодействие пользователей с веб-технологиями стало повсеместным: от заказа еды и покупок в интернет-магазинах до управления корпоративными процессами и аналитики больших данных.

Кроме того, веб-разработка активно влияет на развитие искусственного интеллекта (Дукаев, 2024), облачных технологий (Ступина, 2023) и Интернета вещей (Якименко, 2017). Современные веб-приложения интегрируют алгоритмы машинного обучения, автоматизируя процессы обработки информации, персонализации контента и предсказания пользовательских предпочтений. Это позволяет компаниям более эффективно работать с данными, улучшая качество услуг и удобство взаимодействия.

С точки зрения технической реализации, веб-разработка включает в себя широкий спектр инструментов и технологий, таких как языки программирования (JavaScript, Python, PHP), фреймворки (React, Angular, Vue.js), базы данных (MySQL, PostgreSQL, MongoDB) и системы управления контентом (WordPress, Joomla, Drupal). Каждый из этих элементов играет важную роль в создании функциональных, удобных и безопасных веб-ресурсов.

Еще совсем недавно, в начале 2000-х годов, веб-разработка воспринималась как некая дополнительная функция для рекламных целей в маркетинговой деятельности компаний, но с развитием Интернета и цифровых технологий её значение существенно выросло. В наши дни веб-присутствие стало необходимостью практически для любого бизнеса, будь то небольшой магазин или крупная корпорация (Петров, 2019). Веб-сайт является лицом компании в онлайн-мире, его качество и функциональность напрямую влияют на впечатление, которое та или иная организация старается произвести на своих потенциальных клиентов. Веб-разработка прошла долгий путь от простых статических страниц к динамическим и интерактивным веб-приложениям. В самом начале своего развития, с появлением Интернета, веб-сайты состояли в основном из текста и изображений, а взаимодействие с пользователями было ограничено (там же). Однако с развитием информационно-компьютерных технологий (ИКТ) появились новые возможности, такие как динамическое обновление содержимого, анимации, мультимедийные элементы, интерактивные формы и другие функции, делающие веб-сайты более привлекательными и функциональными для пользователей (там же).

С появлением мобильных устройств и различных устройств для доступа к Интернету веб-разработка стала более разнообразной и адаптивной (Сергачева, Михалевская, 2020, с. 35). Сегодня веб-разработчики должны учитывать размеры экранов и характеристики устройств в целях обеспечения оптимального пользовательского опыта на всех платформах. Кроме того, в связи с увеличением использования аналитики и персонализации контента для улучшения пользовательского опыта и повышения эффективности веб-сайтов веб-разработка стала более ориентированной на пользователя. Веб-разработка стала значительно более сложной и многосторонней по сравнению с тем, какой она была несколько десятилетий назад. Раньше веб-разработчикам для создания статических страниц (то есть веб-страниц, содержимое которых остается неизменным и не зависит от пользовательских действий или внешних данных) требовалось преимущественно

владение HTML и CSS, в то время как сегодняшним специалистам в области веб-разработки необходимо обладать глубокими знаниями в сферах серверных технологий и владеть практическими навыками работы с языками программирования, серверными технологиями, базами данных, а также понимать принципы дизайна и пользовательского опыта (Куваев, Антимонов, 2020, с. 1270).

Сложность задач веб-разработчика увеличилась в наши дни из-за требований к созданию интерактивных и адаптивных веб-приложений, которые должны быть доступны на различных устройствах и в различных сценариях использования (Бондаренко, 2018, с. 46). Это требует понимания потребностей и предпочтений пользователей, а также технических особенностей каждой платформы.

Как уже было сказано выше, сегодня разработка веб-страницы – комплексный процесс, включающий в себя множество этапов. К таким этапам относятся:

- проектирование WEB-приложения или самого сайта, т.е. сбор и последующий анализ всех требований к проекту, разработка технического задания, составление плана проекта для определения архитектуры системы, выбора технологий и инструментов разработки, а также распределения задач между участниками команды;

- выработка концепции сайта с учётом творческого подхода. Сюда входят выбор цветовой палитры, типографики, визуальных эффектов, а также ключевых особенностей пользовательского интерфейса;

- разработка дизайнерской концепции Интернет-ресурса;

- разработка макетов страниц сайта;

- создание и выполнение «FLASH-элементов» (мультимедийных объектов, созданных с использованием технологии Adobe Flash) и мультимедиа;

- верстание шаблонов и страниц;

- работы по программному обеспечению веб-разработки, например, создание функциональных инструментов и их интеграция в уже существующую систему;

- размещение разработанной программы на сайте и оптимизация его текстовых материалов;

- тестирование сайта и внесение, при необходимости, корректировок;

- запуск созданного проекта на общественной площадке в сети Интернет;

- работы по обслуживанию уже действующего портала или его программной части (Власова, 2021; Матвеева, 2019).

Таким образом, этапы веб-разработки охватывают целый ряд ключевых процессов, начиная от создания дизайна сайта и заканчивая его размещением на сервере для публичного доступа. При этом каждый этап тесно связан с другими, и в зависимости от конкретной задачи могут применяться различные подходы и методики к разработке веб-сайта. Начальным этапом процесса разработки веб-сайта является создание дизайна, который обычно осуществляется с использованием графического редактора. Здесь важно учитывать технические ограничения стандарта HTML, чтобы созданный дизайн можно было успешно реализовать в виде веб-страниц (Власова, 2021).

После утверждения дизайна веб-страницы заказчиком специалист по верстке разбивает графическое изображение на отдельные элементы и создает HTML-страницы, используя созданный дизайн в качестве основы. Эти страницы могут быть реализованы как статически, так и с использованием CMS (англ. content management systems – систем управления контентом).

Далее следует этап разработки программного обеспечения веб-сайта, во время которого программист обеспечивает функциональность сайта в соответствии с заданными требованиями. Этот процесс может включать как написание кода «с нуля», так и использование готовых CMS и их настройку под нужды проекта (там же).

На финальном этапе веб-сайта производится его тестирование и оптимизация. Это включает в себя проверку работоспособности сайта в различных браузерах и на

разных устройствах, исправление обнаруженных ошибок, а также оптимизацию производительности веб-приложения и его безопасности.

Завершающим этапом разработки веб-сайта является размещение файлов сайта на сервере и его настройка для обеспечения Интернет-доступа. После этого сайт становится доступным для публичного использования, хотя некоторые настройки могут продолжаться и после реализации данного этапа (там же).

Основными языками, используемыми для создания визуальной составляющей веб-страницы, являются HTML, CSS и JavaScript (Сергачева, Михалевская, 2020, с. 35; Двуреченский, 2023, с. 42). Каждый из этих языков выполняет свою функцию в формировании веб-страницы, предоставляя разработчикам возможность создавать уникальные и интерактивные пользовательские интерфейсы.

HTML является основным языком разметки, который определяет структуру содержимого веб-страницы (Матвеева, 2019, с. 537). С помощью HTML веб-разработчики указывают, какие элементы присутствуют на странице и как они организованы. Например, с помощью тегов и <p> можно создавать блоки текста, а теги <img> и <video> используются для вставки изображений и видео. HTML обеспечивает базовый каркас страницы, определяя её структуру, но не управляет её внешним видом или поведением.

Для стилизации и визуального оформления веб-страниц используется CSS. Каскадные таблицы стилей позволяют разработчикам определять внешний вид элементов, таких как цвет, шрифты, размеры и расположение на странице (Злобин, Скворцова, 2023, с. 56). Они обеспечивают возможность создания современного дизайна, который обеспечивает визуальную привлекательность веб-страницы.

JavaScript является языком программирования, который придаёт веб-страницам интерактивность и динамизм (Ершов, Голубничий, 2023). С его помощью можно создавать различные эффекты, анимации, обрабатывать события пользователя и взаимодействовать с сервером без перезагрузки страницы. JavaScript позволяет создавать сложные функциональные элементы, такие как слайдеры, выпадающие

меню и формы обратной связи, что повышает удобство использования и функциональность веб-страницы. Благодаря языку программирования JavaScript можно создавать более динамичные и привлекательные интерфейсы, улучшая тем самым пользовательский опыт.

Помимо этого, веб-разработчики используют различные фреймворки для ускорения процесса создания веб-приложений и облегчения собственной работы. Поясним, что фреймворк – это некая основа или набор инструментов, предназначенный для разработки веб-приложений (Сергачева, Михалевская, 2020, с. 35). Он предоставляет разработчикам готовые компоненты, библиотеки и структуры, которые помогают создавать функциональность веб-приложений с меньшими усилиями.

Некоторые из основных фреймворков веб-разработки включают в себя (Сергачева, Михалевская, 2020, с. 35–37):

React.js представляет собой фреймворк в JavaScript, который используется для создания пользовательских интерфейсов. React позволяет разработчикам создавать компоненты, которые автоматически обновляются при изменении данных, что делает веб-приложения более динамичными и «отзывчивыми» (Бондаренко, 2018, с. 46).

Angular является еще одним популярным фреймворком в JavaScript, который предназначен для создания одностраничных приложений. Angular предоставляет широкий набор инструментов для работы с пользовательским интерфейсом и обработкой данных (Гек, Тихоненко, 2021, с. 486).

Vue.js используется для создания пользовательских интерфейсов, но отличается от React и Angular своей простотой и легкостью использования. Vue.js обеспечивает быструю разработку веб-приложений и хорошо масштабируется для больших проектов, т.е. может эффективно адаптироваться к увеличению объёма данных, пользователей и функциональности без существенных потерь в

производительности. Это позволяет использовать Vue.js как для небольших сайтов, так и для сложных корпоративных веб-приложений (Яровая, 2022, с. 2783).

Express.js предназначен для создания серверных приложений на языке JavaScript. Express.js позволяет разработчикам создавать API (Application Programming Interface, набор правил и инструкций, которые определяют, как различные программные компоненты могут взаимодействовать друг с другом (Абсалямова, Сафиканова, 2023, с. 19)) и обрабатывать HTTP-запросы, что делает его отличным выбором для создания веб-серверов и микросервисов, т.е. архитектуру программного обеспечения, при которой приложение разбивается на небольшие, независимые сервисы, каждый из которых выполняет свою специфическую задачу.

Следует отметить, что ко всем вышеупомянутым технологиям по веб-разработке предоставляется подробная документация от разработчика. Эта документация содержит информацию о том, как использовать технологию, какие функции она предоставляет, а также примеры кода и руководства по реализации различных задач. Как было отмечено выше, документация играет важную роль в процессе разработки, обеспечивая разработчиков необходимой информацией для успешной работы с каждой конкретной веб-технологией. В следующем подразделе будут рассмотрены классификации текстов в IT-сфере и определено место официальной документации в области веб-разработки в системе представляемых текстов.

1.4.1. Техническая документация как разновидность технических текстов

Технические тексты представляют собой разновидность научно-технического подстиля языка, в котором функциональная направленность и узкая предметно-тематическая специфика выходят на первый план. В составе любого текста такого типа встречаются профессиональные жаргонизмы, отсутствующие в общеязыковых словарях, что подчёркивает его ориентированность на профессиональное сообщество и невозможность полного понимания неспециалистом (Нишонов, 2020, с. 47). Согласно мнению А.Ю. Алипичева, А.Н. Кузнецова и Н.А. Сергеевой, технические тексты принципиально отличаются от художественных отсутствием метафор, выразительного авторского стиля и наличием чётко очерченной узкой предметной направленности, что отвечает прежде всего их инструктивной и регламентационной роли (Алипичев, 2024, с. 83). Технические тексты могут включать описания сложнейших технологических процессов, содержать таблицы, графики и схемы, а также отсылки к понятиям, не раскрываемым непосредственно в тексте, – всё это требует от читателя определённого уровня профессиональной подготовки и контекстной осведомлённости (Авцынова, 2023, с. 796; McCarthy, 2021, p. 199).

Как отмечает Л. Г. Федюченко, технический язык можно уподобить диалекту внутри более широкого языкового сообщества: особая лексика, синтаксис и прагматика такого «диалекта» формируются спецификой технической сферы и нередко оказываются малопонятны вне узкой группы специалистов (Федюченко, 2019, с. 293). Несмотря на отсутствие единого общепринятого определения технического текста, в отечественной лингвистике сложились близкие по смыслу трактовки. Так, А. Л. Буран определяет технический текст как «особую единицу коммуникации с единой системой терминологии, имеющей однозначное толкование и понимание среди специалистов определённой профессиональной сферы» (Буран, 2012, с. 97). Данное определение представляется методологически важным, поскольку фиксирует два базовых признака технического текста: его коммуникативную природу и терминологическую организованность.

А. Ю. Горбунов и Л. А. Долбунова, в свою очередь, подчёркивают такие свойства технического текста, как письменный и преимущественно монологический характер, строгий отбор языковых средств, нормированность речи, логическая последовательность, а также стремление к точности, сжатости и однозначности при сохранении высокой информационной насыщенности (Горбунов, Долбунова, 2015).

Таким образом, в формулировании рабочего определения технического текста целесообразно опереться на точку зрения А. Л. Буран, которая позволяет наиболее чётко зафиксировать сущность технического текста как особой единицы профессиональной коммуникации. Вместе с этим требуется уточнение признаков, которые выделяются в других исследованиях, а именно письменного и преимущественно монологического характера технического текста, его логической последовательности, нормированности, точности, сжатости, однозначности, высокой информационной насыщенности и выраженной инструктивно-регламентационной направленности. С учётом данных характеристик технический текст в данном исследовании понимается как особая единица профессиональной коммуникации, основанная на системе терминов и ориентированная на точную, логически последовательную и функционально значимую передачу технического знания в пределах определённой профессиональной сферы.

1.4.1.1 Основные характерные черты технических текстов

Технические тексты обладают специфическими чертами, обеспечивающими ясную и точную передачу информации в научно-технической и производственной сферах. Они предназначены для профессионалов, которым важно точно

воспроизводить содержание в своей работе (Федюченко, 2019, с. 294). Особенности технических текстов касаются как лексических, так и грамматических аспектов, а также структуры и подачи информации. Рассмотрим данные особенности.

Прежде всего, следует отметить, что технические тексты, как правило, ориентированы на передачу объективных данных и не включают субъективную оценку. Л.А. Коняева утверждает, что тексты научно-технической направленности характеризуются «точностью, безличностью и отсутствием эмоциональной окрашенности» (Коняева, 2015, с. 52). Безличность особенно важна в контексте точности и объективности: технический текст должен быть лишён какого-либо влияния субъективного мнения автора. Использование безличных конструкций в технических текстах помогает добиться эффекта субъективности, поскольку такие конструкции позволяют устранить присутствие автора и исключают наличие эмоциональных оценок (Rodić, 2025). Таким образом, технические тексты ориентированы на передачу объективных сведений, а не личных взглядов и предпочтений.

С.С. Калинина подчеркивает, что научный стиль, частью которого и являются технические тексты, обладает рядом общих признаков, таких как «абстрактность, отвлеченность, обобщенность, логичность и смысловая точность материала» (Калинина, 2017, с. 66). Этим технические тексты отличают от других жанров научной литературы. В отличие от текстов, принадлежащих гуманитарным наукам, где возможны элементы публицистического стиля, технические тексты стремятся к максимальной конкретике и однозначности. Они исключают двусмысленность, которая могла бы помешать точной передаче информации.

Следует согласиться с С.С. Калининой в том, что неотъемлемой частью структуры технических текстов являются стандартные языковые конструкции и

клише. Такие клише помогают упорядочить информацию и обеспечить её логическое построение (там же).

Важным аспектом создания технических текстов является стандартизация. Вопрос о стандартизации касается прежде всего лексических средств. Для этого типа текста характерен специфический набор терминов и определений, которые способствуют точной передаче информации (там же). Стандартизация облегчает восприятие текста профессионалами, поскольку они могут быстро ориентироваться в нём и находить необходимую информацию. Как отмечают Е.А. Сидорова и соавторы, техническая документация играет важную роль при проектировании и разработке сложных систем, таких как программные комплексы (Сидорова и др., 2018, с. 30). Любая неточность в формулировках или недостаточная детализация может привести к снижению качества конечного продукта. Поэтому стандартизация в написании технических текстов необходима для того, чтобы избежать ошибок, которые могут возникнуть при интерпретации технических требований.

Как отмечает Л.А. Коняева, технические тексты отличает «формально-логический стиль», в котором используется множество специализированных терминов, в том числе латинской и греческой этимологии (Коняева, 2015, с. 52). Эти термины играют ключевую роль в построении текста, так как они являются основным средством передачи знаний. Подбор лексики в рассматриваемых текстах особенно важен, поскольку от правильного выбора терминов зависит точность передачи информации. Л.А. Коняева подчёркивает, что в технических текстах широко используются служебные слова и слова, обеспечивающие логические связи между элементами описания (там же). Это важно для создания связного и логически построенного текста.

Слово «точность» в контексте технических текстов можно понимать как отражение требований профессиональных стандартов. В технических текстах авторы

стремятся исключить любые возможные ошибки или двусмысленности, которые могут возникнуть при интерпретации текста. Как отмечает В. Н. Базылев, терминология в технической литературе должна быть однозначной, чтобы исключить возможность неверного понимания информации (Основные понятия переводоведения…, 2010, с. 198). Лексический состав технических текстов включает слова и словосочетания, обозначающие конкретные или абстрактные понятия в той или иной области техники: это могут быть инструменты, приборы, машины, рабочие операции или единицы измерения (там же).

Как указывает Д.В. Семенов, терминология является одним из важнейших компонентов технического текста, поскольку она обеспечивает точное и недвусмысленное понимание читателем представленной информации (Семенов, 2023, с. 809). При этом технические тексты не углубляются в теоретические аспекты, на которых основан предмет или процесс, а фокусируются на практическом применении знаний. Важно отметить, что основная цель технического текста – передача когнитивной информации, которая должна быть объективной и легко интерпретируемой (Семенов, 2023, с. 810). Для достижения этого ученые используют ряд специальных языковых средств, включая номинативные конструкции и пассивный залог, что способствует формированию параметров абстрактности и объективности.

Говоря о грамматике технических текстов, следует отметить, что для технического стиля характерны сложные синтаксические конструкции, что подчёркивает логичность и последовательность изложения. Я.И. Рецкер отмечает, что в технической литературе преобладают сложносочиненные и сложноподчинённые предложения, в которых используется большое количество существительных, прилагательных и неличных форм глагола (Рецкер, 2009, с. 67). Это создаёт текст, который по своей структуре напоминает формулу или модель, где

каждая часть выполняет определённую функцию. Более того, использование пассивных конструкций позволяет автору сосредоточить внимание на предмете исследования, а не на субъекте действия.

В дополнение ко всему вышесказанному, следует отметить, что в технических текстах большое внимание уделяется визуализации данных. Использование графиков, таблиц, схем и чертежей является важным средством представления информации. Это особенно актуально для технической литературы, которая нередко содержит сложные для восприятия числовые данные и модели. Как указывает С.С. Калинина, графики и диаграммы являются важной частью технических текстов, поскольку они позволяют представить сложные концепции в более доступной форме (Калинина, 2017, с. 66). Это делает информацию более понятной для специалистов, работающих в конкретной области.

И наконец, ещё одной важной характерной чертой технических текстов является их номинативный стиль. Как подчеркивает Д.В. Семенов, в техническом стиле доминируют существительные, а действие выражается преимущественно через отглагольные существительные, что увеличивает уровень абстракции текста [Семенов 2023: 810]. Это позволяет автору избежать личных оценок и сосредоточиться на объективной передаче информации.

Таким образом, технический текст представляет собой особую разновидность научно-технического текста, для которой характерны объективность, безличность, логическая последовательность, стандартизованность, терминологическая насыщенность, стремление к точности, однозначности и сжатости изложения. Существенную роль в его организации играют клишированные конструкции, сложный синтаксис, номинативный характер изложения и использование визуальных средств представления информации. Совокупность этих черт обеспечивает эффективную передачу специального знания в профессиональной

среде и делает технический текст функционально ориентированным на точное восприятие и практическое применение содержащихся в нём сведений.

Отдельный вид технических текстов, обладающий своими структурными и языковыми особенностями, представляют собой тексты технической документации, в том числе документации в области веб-разработки.

1.4.1.2 Характеристики технической документации в области веб-разработки

Документация в области веб-разработки представляет собой специализированный вид технических текстов, направленный на подробное описание функционала и возможностей той или иной технологии, используемой в разработке веб-приложений (Meng, 2019; Fan, 2020). Основные задачи, которые ставятся перед такими текстами, включают:

1) объяснение функциональных возможностей новой технологии,

2) предоставление точных инструкций по её использованию,

3) демонстрация её преимуществ по сравнению с альтернативами,

4) обучение разработчиков эффективному применению данной технологии в практических ситуациях (González-Mora, 2023).

Документация по веб-разработке обладает признаками интернет-текста, поскольку существует в электронной форме, размещается в сетевой среде, имеет гипертекстовую организацию и включает вербальные и невербальные компоненты. Л. Ю. Щипицина относит к лингвистически релевантным параметрам компьютерно-

опосредованной коммуникации электронную форму существования текста, гипертекстуальность, мультимедийность, интерактивность, синхронность, а также количество и эксплицированность коммуникантов (Щипицина, 2010). Эти характеристики позволяют рассматривать документацию по веб-разработке как интернет-текст, совмещающий признаки технического и учебного текста. Инструктивный характер документации по веб-разработке соотносится с общеязыковым пониманием инструкции как «свода правил, устанавливающего порядок и способ осуществления чего-либо», а также как «руководящих указаний или наставлений» (Большой толковый словарь русского языка, 2000).

Таким образом, документация по веб-разработке призвана знакомить разработчиков с теоретическими аспектами новой технологии и обеспечивать их практическими инструментами для решения конкретных задач, возникающих в процессе работы.

Возникновение текстов документации по веб-разработке тесно связано с развитием и появлением новых технологий в сфере веб-разработки (там же). Поскольку эта область отличается высокой динамичностью и постоянным обновлением, в условиях непрерывного совершенствования технологий, появления новых инструментов тексты документации становятся необходимостью. Веб-разработчики часто сталкиваются с необходимостью осваивать совершенно новые подходы и языки программирования, а также инструменты, которые существенно меняют процессы разработки новых веб-приложений (там же). Сюда можно отнести появление новых библиотек языка программирования JavaScript, таких как React JS или Vue.js, которые потребовали создания подробной документации для их освоения разработчиками. Поскольку рынок веб-разработки постоянно развивается, успешная интеграция новых технологий требует доступных и эффективных обучающих материалов, представленных в виде технической документации.

Ключевой особенностью такой документации является её структурированность и логическая организация материала. Основное содержание документации по веб-разработке направлено на описание каждого функционального элемента технологии. Например, документация по фреймворку Angular включает подробные разделы, посвящённые каждой из его составляющих – модулям, компонентам и сервисам (Angular, заглавная страница). Кроме того, для каждой функциональной части обычно предоставляются примеры кода, иллюстрирующие использование описываемой технологии (см. Рисунок 1). Такие примеры могут варьироваться от простых базовых вариантов до более сложных сценариев, применяемых в реальных проектах. Примеры кода играют важную роль в восприятии материала, поскольку они помогают разработчикам быстрее освоить теоретическую информацию и применить её на практике. На эффективность таких примеров указывают многочисленные исследования (Hoffman, 2003; Zhang, 2021; Bogner, 2023; Allen, 2023).

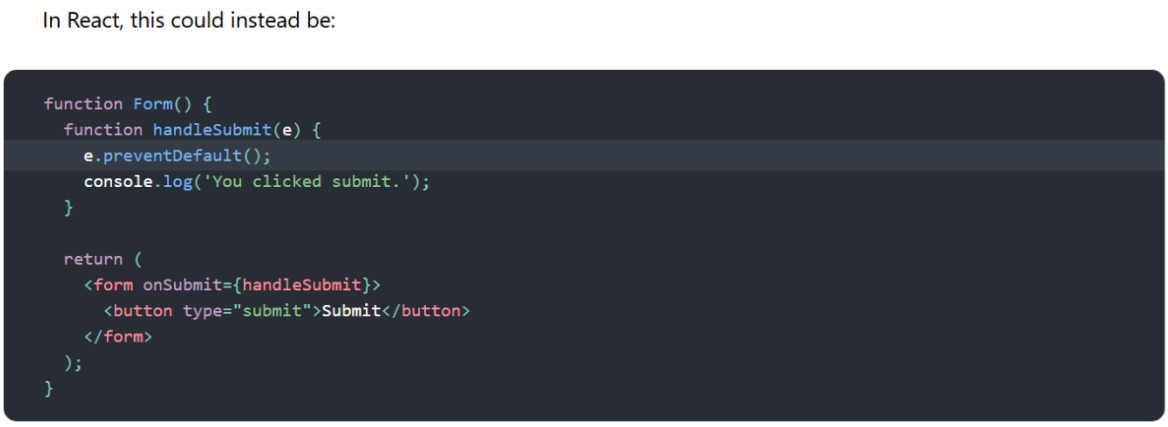


Рисунок 1. Интеграция примера кода в тексте документации к React.js

Следующим важным элементом текстов документации по веб-разработки является описание типичных ситуаций использования технологии. В текстах документации, как правило, указываются примеры сценариев, в которых данная технология наиболее эффективна. Например, в документации по библиотеке React

могут описываться примеры создания одностраничных приложений с подробным объяснением того, как компоненты взаимодействуют друг с другом (Allen, 2023). Такие примеры помогают разработчикам понять, в каких конкретных ситуациях стоит применять ту или иную технологию. Кроме того, рассматриваются типичные ошибки, которые могут возникнуть в процессе работы, и способы их устранения. Это даёт пользователю общее представление о технологии, а также предупреждает о возможных проблемах (там же).

Документация также уделяет значительное внимание преимуществам новой технологии. Одной из задач документации является стремление убедить разработчика в том, что предложенная технология решает существующие проблемы лучше, чем её предшественники или конкуренты (Myllärniemi, 2018). Например, документация по фреймворку Vue.js подчёркивает его простоту и гибкость в сравнении с более сложными решениями, такими как Angular (Vue.js, руководство по применению). В таких текстах обычно акцентируется внимание на производительности, удобстве использования, простоте интеграции с другими инструментами и гибкости в настройке. Эти аргументы сопровождаются сравнительными таблицами и статистикой, что позволяет пользователю наглядно увидеть преимущества новой технологии. Это представляет особую ценность, поскольку разработчики часто сталкиваются с выбором между несколькими инструментами, а документация помогает им сделать осознанный выбор.

На основе опыта работы с документацией к технологиям по веб-разработке можно выделить несколько её структурных особенностей. Во-первых, тексты документации по веб-разработке, как правило, разделены на несколько основных разделов: введение, руководство по началу работы, описание основных функций и возможностей, справочник API и раздел с частыми вопросами. Введение и руководство по началу работы с очередной технологией веб-разработки содержат

базовую информацию, необходимую для начала работы с технологией, такие как установка, настройка окружения и создание первого простого проекта. Здесь также можно встретить раздел с часто задаваемыми вопросами, который помогает решать наиболее распространённые проблемы и вопросы, которые могут возникнуть у пользователей.

Во-вторых, важным элементом технической документации к технологиям веб-разработки являются разделы с примерами и упражнениями. Это позволяет пользователям подобной технической документации применять полученные знания на практике, выполняя небольшие задания. В документации по JavaScript-библиотеке D3.js, например, большое внимание уделено визуализации данных, и на основе предоставленных примеров разработчики могут создавать собственные интерактивные графики (D3.js, руководство по применению). Такие практические задания позволяют закрепить знания и лучше понять, как технология работает в реальных условиях.

В-третьих, ещё одной характерной чертой документации по веб-разработке является её вариативность. Тексты документации могут отличаться друг от друга, так как не существует строгих, общепринятых правил её составления. В отличие от некоторых других типов технических текстов, где стандарты и форматы строго регламентированы, техническая документация к технологиям по веб-разработке составляется на усмотрение разработчика или команды разработчиков. Это приводит к тому, что каждая такая документация имеет свой стиль, структуру и язык.

В-четвёртых, документация по технологиям веб-разработки характеризуется также доступностью языка. Это проявляется в использовании простого и понятного языка, особенно в разделах, предназначенных для начинающих пользователей. Хотя тексты насыщены техническими терминами, общий стиль изложения остаётся доступным для понимания даже теми, кто только начинает знакомиться с

технологией. Например, в руководствах по работе с HTML и CSS (напр., MDN Web Docs) часто встречаются подробные объяснения базовых понятий, таких как теги или стили, с указанием примеров кода и визуальных примеров. Эта простота обусловлена тем, что документация ориентирована на широкую аудиторию, включающую как опытных разработчиков, так и новичков, которым важно быстро освоить новый инструмент.

Важной частью текста документации по веб-разработке являются многочисленные пояснения и ссылки на сторонние ресурсы. Такие тексты часто содержат ссылки на другие источники информации в виде гиперссылок, таких как форумы разработчиков, блоги или дополнительные материалы по теме. Например, в официальной документации React JS при объяснении ключевого слова *export default* пользователю предлагается перейти по гиперссылке на ресурсы MDN Web Docs и javascript.info (см. Рисунок 2), где можно подробнее узнать о синтаксисе JavaScript. Таким образом, текст предоставляет путь к более фундаментальному пониманию темы. Такая стратегия особенно эффективна при работе с начинающими разработчиками, которым важно сразу получить доступ к доступно изложенному базовому материалу, подкреплённому примерами и визуализацией кода.

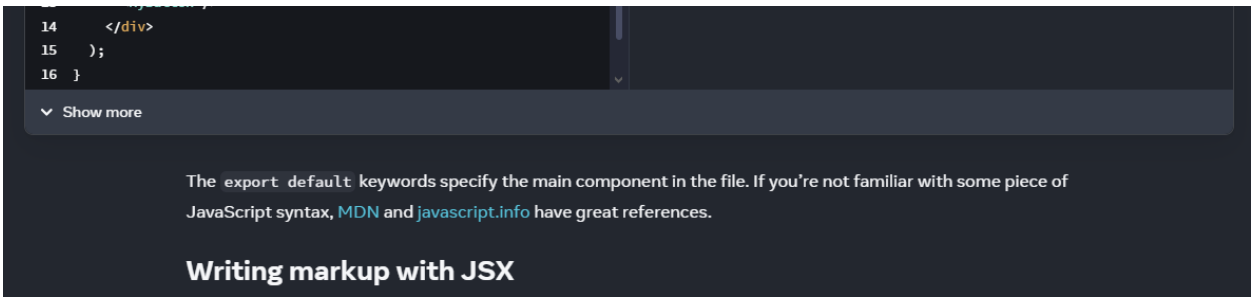


Рисунок 2. Пример интеграции гиперссылки на сторонние ресурсы в документации к React.js

Относительно простой язык, характерный для документации по веб-разработке, обусловлен её функциональной направленностью. Основная задача таких текстов – обеспечить разработчиков необходимыми знаниями для работы с определённой технологией. В связи с этим документация по веб-разработке

составляется с учётом того, что аудитория должна быстро усваивать информацию и применять её на практике. Это влияет на стиль и язык изложения: тексты рассматриваемого типа документации избегают метафор, сложных стилистических конструкций и избыточной терминологии, которые могут затруднить понимание. Вместо этого используются чёткие и лаконичные формулировки, а также термины, которые однозначно интерпретируются в ИТ-сфере.

В некоторых случаях тексты документации по веб-разработке могут включать элементы визуальной информации: диаграммы (Рисунок 3), схемы (Рисунок 4), таблицы (Рисунок 5).

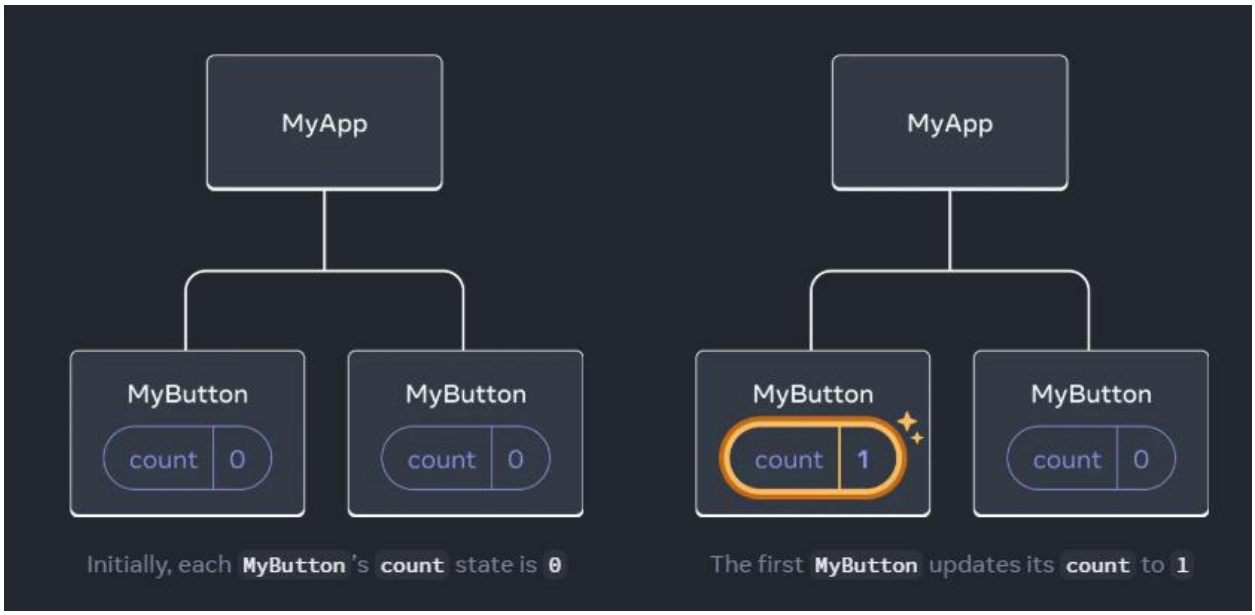


Рисунок 3. Диаграмма, демонстрирующая различные состояния компонентов в документации к React.js

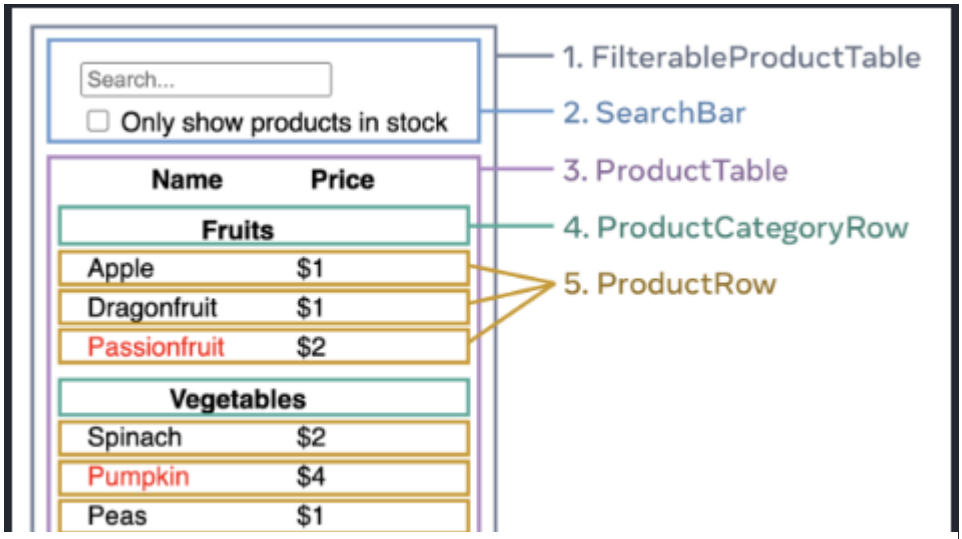


Рисунок 4. Схема компонентного взаимодействия в React JS

Syntax	Export statement	Import statement
Default	`export default function Button() {}`	`import Button from './Button.js';`
Named	`export function Button() {}`	`import { Button } from './Button.js';`

Рисунок 5. Сравнение синтаксиса экспортных и импортных операторов в JavaScript: default и named export в документации к React.js

Таким образом, документация по веб-разработке представляет собой особую разновидность технической документации, сочетающую справочную, инструктивную и обучающую функции. Её отличают структурированность, ориентация на практическое освоение технологии, активное использование примеров кода, гиперссылок, визуальных элементов и сценариев применения, а также стремление сделать новый инструмент понятным и функционально доступным для разработчика. В условиях постоянного обновления веб-технологий такая документация становится неотъемлемым средством их освоения и внедрения, обеспечивая трансфер специального знания в профессиональную среду.

1.4.2. Особенности перевода текстов документации в области веб-разработки

Перевод технических текстов, к которым относятся тексты документации к технологиям веб-разработки, представляет собой сложную и многослойную задачу, требующую от переводчика глубокого знания языков и умения ориентироваться в специфике отрасли, терминологии и контексте. Основная сложность перевода технических текстов заключается в необходимости передавать содержание

максимально точно, избегая малейших погрешностей, которые могут существенно изменить смысл текста. Как отмечает Л.А. Коняева, малейшие погрешности перевода могут изменить смысловую нагрузку текста (Коняева, 2015), что делает перевод технической литературы особо ответственным видом деятельности.

Прежде всего, при переводе технических текстов необходимо учитывать такой фактор, как междисциплинарность перевода технических текстов. По мнению А. Л. Пумпянского, «перевод научной и технической литературы является особой дисциплиной, возникшей на стыке лингвистики, с одной стороны, и науки, и техники – с другой» (Пумпянский, 1965, с. 21). Это утверждение подчеркивает важность междисциплинарного подхода, согласно которому языковая точность не должна нарушать научную и техническую точность текста. Для достижения адекватности перевода необходимо одновременно учитывать как лингвистические, так и технические аспекты.

Л. Л. Нелюбин связывает такой перевод с значительной терминологичностью, предельно точным и чётким изложением материала, а также с необходимостью владения терминологией взаимодействующих языков и умением правильно пользоваться ей (Нелюбин, 2003, с. 117). Развивая эту мысль, Л. А. Коняева подчеркивает, что профессиональный технический перевод возможен только при наличии у переводчика глубоких знаний предметной области и специфической терминологии (Коняева, 2015, с. 51). Кроме того, переводчик должен обладать высоким уровнем лингвистической подготовки, что включает в себя знание иностранных языков и умение грамотно передавать свои мысли на языке перевода, сохраняя при этом стиль и содержание оригинала.

Одной из главных характеристик, определяющих профессионализм переводчика в области технического перевода, является владение как общей, так и специальной терминологией, в нашем случае технического характера. Как отмечает

В.Н. Базылев, «техническая терминология – это слова или словосочетания, обозначающие конкретные или абстрактные понятия той или иной области техники» (Основные понятия переводоведения…, 2010, с. 199). Термины часто однозначны в пределах своей профессиональной области, что делает знание этой терминологии критически важным для адекватного перевода (Котова, 2013, с. 134). Однако, как подчеркивают ученые, «эта однозначность относительна», и значение термина может варьироваться в зависимости от контекста его употребления (Основные понятия переводоведения…, 2010, с. 199; Тиллаева, 2019, с. 15). Например, переводчик должен тщательно выбирать то или иное значение многозначного термина, основываясь на конкретном контексте в определенной отрасли знаний, хотя иногда такой контекст может отсутствовать, что создает дополнительные трудности. Это особенно заметно при передаче в переводе списков или таблиц, в которых термины приводятся в изолированной форме, без явного контекста, что усложняет выбор точного эквивалента. Усугубляет ситуацию и тот факт, что технические термины, по словам В. Н. Базылева, являются наиболее динамичным слоем лексики, который постоянно обновляется в связи с научно-техническим прогрессом (Основные понятия переводоведения…, 2010, с. 199). Это означает, что переводчик должен постоянно быть в курсе новейших достижений науки и техники, чтобы уметь правильно переводить новые термины и понятия.

В качестве одного из примеров однозначных терминов в документации к технологиям веб-разработки можно привести лексическую единицу *props.* В контексте фреймворка React JS данный термин представляет собой сокращение от лексемы *properties* и обозначает механизм передачи данных между компонентами приложения. Важной особенностью термина *props* является его неизменяемость: родительский компонент передает свойства (т.е. props) дочернему, но сам дочерний компонент не может их изменять, а только использовать (Хайруллин, Богданов,

2024). Это фундаментальный принцип работы React JS, который поддерживает концепцию однонаправленного потока данных.

При переводе документации важно сохранять точность термина и избегать возможных искажений. Дословный перевод терминологической единицы *props* как «свойство» может быть технически корректным, но он не передает всей специфики рассматриваемого понятия в контексте React JS. В русскоязычной документации данный термин часто оставляют без перевода, как *props*, чтобы разработчики могли напрямую соотносить его с оригинальными материалами. Однако в пояснительном тексте можно было бы дать уточнить его как «передаваемые параметры» или «параметры компонента» для того, чтобы можно было подчеркнуть его функцию в архитектуре React-приложения.

В переводе технической документации по веб-разработке используются термины, которые могут иметь несколько значений в зависимости от контекста. Одним из таких многозначных терминов является лексема *component*. В технической документации данный термин может обозначать как функциональные и классовые компоненты, которые являются основными строительными единицами React-приложений, так и элементы пользовательского интерфейса, которые могут быть частью более сложных структур (Хайруллин, Богданов, 2024). Кроме того, термин *component* иногда используется в текстах по веб-разработке и в архитектурном смысле, обозначая отдельные модули системы, например, компоненты маршрутизации или управления состоянием. При переводе документации по веб-разработке важно учитывать контекст и адекватно интерпретировать значение термина. Если речь идет о React JS, корректным переводом будет русскоязычный термин «компонент», однако, если термин используется в более общем смысле, может потребоваться уточнение, например, «элемент интерфейса» или «модуль системы».

Еще одним многозначным термином, часто встречающимся в документации React JS, является лексема *state.* В зависимости от контекста данный термин может означать состояние динамических данных, которые управляют его поведением и отображением. В другом контексте термин *state* может обозначать глобальное состояние приложения, хранящегося в централизованном хранилище и доступного нескольким компонентам через механизмы управления состоянием. В более широком смысле слова данный термин может также относиться к состоянию системы или веб-приложения, указывая на текущие параметры последнего и его конфигурацию (Бондаренко, 2018, с. 47). При переводе необходимо точно определять значение термина *state*, исходя из контекста использования. Если речь идет о локальном состоянии React-компонента, наиболее адекватным переводом следует считать «состояние компонента», тогда как в случае глобального состояния точнее будет использовать «глобальное состояние» или «хранилище данных».

Особое внимание следует уделять передаче в переводе безэквивалентных терминов, которые не имеют точного аналога в языке перевода и требуют адаптации или пояснения. В текстах по веб-разработке такие термины встречаются довольно часто, поскольку веб-технологии развиваются стремительно, а новые концепции и инструменты появляются быстрее, чем успевают закрепиться устоявшиеся переводы. Например, термин *hook*, используемый в React JS, не имеет точного аналога в русском языке. Он обозначает механизм, позволяющий функциональным компонентам использовать состояние и другие возможности React JS без написания классов (Антонов и др., 2024). В русскоязычной документации данный термин обычно оставляют без перевода, однако при пояснении могут использоваться такие описательные формулировки, как «механизм подключения» или «способ взаимодействия с состоянием».

Еще один пример – термин *serverless*, который в контексте веб-разработки не означает «без сервера» в буквальном смысле, а описывает архитектуру, при которой управление серверами абстрагировано от разработчика, а вычисления выполняются в облаке на основе событий, запускаются автоматически в ответ на определённые действия или изменения, например отправку формы, HTTP-запрос, загрузку файла или изменение данных в базе. В русскоязычной технической литературе используются как транслитерированный вариант «серверлесс», так и описательный перевод, например «бесперебойное облачное выполнение» или «функции как сервис». Адаптация таких терминов для русскоязычных разработчиков играет важную роль в переводе документации, поскольку помогает разработчикам адекватно интерпретировать новый для себя термин, не создавая ложных представлений.

В тех случаях, когда некоторое явление присутствует в обоих языках, но общепринятый перевод еще не сформировался, переводчику приходится либо использовать транслитерацию, либо вводить новые термины, способные точно передать смысл. Это особенно актуально для информационных технологий, где появление новых терминов требует гибкого подхода к переводу, обеспечивающего корректное восприятие и адаптацию терминологии к целевой аудитории.

Примером англоязычного термина является также лексема *cloud*, которая на русский язык переводится как «облако». В контексте веб-разработки и ИТ-индустрии этот термин обозначает распределенные вычислительные ресурсы, доступные через интернет. Облачные технологии позволяют пользователям хранить данные, запускать приложения и использовать вычислительные мощности удаленных серверов, что значительно упрощает управление инфраструктурой и снижает затраты на оборудование. В русскоязычной технической документации термин «облако» широко используется и закрепился как стандартный перевод.

Существуют, однако, и термины, эквиваленты перевода которых еще не устоялись. Например, с помощью терминологической единицы *edge computing* обозначается концепция, предполагающая обработку данных на периферийных узлах сети, ближе к источнику их возникновения, а не в централизованных облачных серверах. На русский язык данный термин иногда переводят как «периферийные вычисления», но окончательный эквивалент пока еще не закрепился. В некоторых случаях для передачи указанного термина в переводе используется транслитерация «эдж-компьютинг». Данная транслитерированная лексическая единица совпадает с оригинальным термином, но не всегда понятна широкой аудитории. Выбор адекватного перевода зависит от контекста и уровня распространенности термина в профессиональной среде.

Специфика передачи таких единиц связана с англоцентризмом терминологии веб-разработки: значительная часть ключевых терминов формируется в англоязычной профессиональной среде и закрепляет не только и не столько техническое обозначение, сколько исходную концептуальную модель технологии. В результате перевод документации предполагает перенос в том числе и способов концептуализации специальных знаний (Вовченко, 2026).

Технический перевод также требует от переводчика умения следовать определенной манере изложения, принятой в той или иной области науки и техники. Как отмечают Е.Д. Полетаева и О.О. Сосновская, в технической сфере «существует устоявшаяся манера изложения», и любое отклонение от неё может быть воспринято как ошибка или непрофессионализм (Полетаева, Сосновская, 2014, с. 2). Это связано прежде всего с тем, что технические тексты, как правило, носят прикладной характер, и их задача заключается прежде всего в передаче четкой и точной информации. Его точность, доступность и логическую согласованность обеспечивает структурно-смысловая адаптация технического текста.

Одним из ключевых аспектов структурно-смысловой адаптации является логическая организация текста. Техническая документация, в том числе и в области веб-разработки, строится по определенным принципам: информация должна быть представлена в строгом порядке, обеспечивающем ее последовательное восприятие. Это включает выделение вводных положений, систематическое изложение материала и логическое завершение, что позволяет читателю эффективно ориентироваться в содержании.

Технические тексты, обладающие высокой степенью формализации, требуют особого подхода, включающего анализ структуры, последовательности аргументации и визуального представления данных (Горбунов, Долбунова, 2015, с. 133). Эти факторы играют решающую роль в передаче технической информации, обеспечивая ее корректное восприятие целевой аудиторией.

Некорректное структурирование технической документации затрудняет восприятие информации и может привести к серьезным ошибкам в применении технологий. Если последовательность изложения не согласуется с логикой предметной области или нарушает ход аргументации, читателю становится сложно проследить авторскую мысль, что усложняет процесс освоения материала и снижает эффективность использования документации. В сфере веб-разработки нарушения в организации текста при переводе могут приводить к неверной интерпретации механизмов работы технологий веб-разработки и, как следствие, вызывать сбои в программных решениях.

Технический перевод требует соблюдения принципов структурно-смысловой адаптации, обеспечивающих последовательность и ясность изложения. В веб-документации особенно важно следовать четко выстроенной логике, которая упорядочивает представление материала и способствует правильному пониманию новых терминов. Документ должен открываться вводными положениями, которые

дают пользователю возможность ознакомиться с ключевыми понятиями и терминологией. Основная часть должна содержать детальное объяснение механизмов, сопровождающееся примерами, а также визуальными элементами, такими как схемы и таблицы, способствующими восприятию сложной информации. Завершающие разделы должны резюмировать изложенное, систематизировать выводы и, при необходимости, предлагать ссылки на дополнительные ресурсы, обеспечивающие углубленное изучение темы.

Строгое соблюдение логической структуры при переводе позволяет избежать искажений и непонимания, поддерживая высокую степень точности передачи информации. В веб-разработке как IT-сфере предъявляются особые требования к качеству документации, поскольку любая неясность в формулировке может привести к ошибкам в коде, неверному использованию API и возникновению проблем в развертывании приложений. Таким образом, структурно-смысловая адаптация делает текст удобочитаемым и играет решающую роль в обеспечении его корректного восприятия разработчиками.

Таким образом, перевод технической документации в сфере веб-разработки требует глубокой вовлеченности в предметную область. От переводчика ожидается понимание логики построения текстов, корректная передача терминологии и способность адаптировать материалы с учетом особенностей профессиональной среды. Однако чисто лингвистических знаний зачастую недостаточно для создания действительно качественного перевода.

Особую сложность представляет работа с многозначными и так называемыми безэквивалентными терминами, а также с новыми концепциями, для которых еще не сформировался общепринятый перевод. Если переводчик не знаком с практическим применением описываемых в технической документации технологий, высока вероятность допущения неточностей, способных исказить смысл документации, что

может привести в свою очередь, к некорректному использованию инструментов разработчиками и, как следствие, к сбоям в работе приложений.

По этой причине идеальным кандидатом на роль переводчика технических материалов является практикующий веб-разработчик. Он обладает языковой и реальным опытом работы с технологиями, понимает принципы их использования и знает нюансы терминологии. Такой специалист интерпретирует текст с учетом контекста использования, исключая риск неверной передачи ключевых понятий. Кроме того, разработчик может адаптировать документацию так, чтобы она соответствовала ожиданиям целевой аудитории.

1.5. Переводческие ошибки и неудачи в контексте перевода технической документации

В рамках настоящего исследования обращение к переводческим ошибкам и неудачам имеет принципиальное значение для выстраивания теоретической основы анализа перевода англоязычной документации по веб-разработке на русский язык. Документация к современным веб-технологиям выступает одним из ключевых каналов трансфера специализированных знаний в профессиональную среду, а перевод, в свою очередь, обеспечивает включение этих знаний в русскоязычный профессиональный дискурс. В связи с этим любые отклонения текста перевода от исходного текста следует рассматривать с позиции сохранения функционального потенциала документации как средства обучения и справочного ресурса.

Анализ переводных текстов показывает, что далеко не все такие отклонения могут быть адекватно описаны через категорию переводческой ошибки. Часть

переводческих решений формально соответствует нормам языка перевода и не содержит явных искажений значения, однако при этом приводит к снижению коммуникативной эффективности текста перевода, изменению его тональности, утрате обучающей направленности или нарушению жанровых ожиданий адресата.

В таких случаях более релевантным оказывается обращение к понятию переводческой неудачи, позволяющему описывать перевод как результат взаимодействия языковых, прагматических и когнитивных факторов, а не исключительно как объект нормативной оценки.

В данном параграфе последовательно рассматриваются традиционные подходы к пониманию переводческой ошибки, вводится и уточняется понятие переводческой неудачи, проводится разграничение этих двух взаимосвязанных понятий, а также обосновывается авторская классификация переводческих неудач применительно к текстам документации по веб-разработке.

1.5.1. Переводческая ошибка как категория переводоведения

Понятие переводческой ошибки занимает место в теории оценки качества перевода как одна из ключевых концепций, с помощью которой описываются отклонения переводного текста от исходного. В отечественной переводоведческой традиции термин «переводческая ошибка» выступает описательной и диагностической категорией, фиксирующей несоответствия в переводе, возникающие при переносе языковых и содержательных элементов исходного текста на язык перевода.

В обзорной статье по переводческим ошибкам в терминологическом словаре-справочнике «Основные понятия переводоведения (отечественный опыт)» отмечается, что переводческие ошибки традиционно определяются как отклонения от исходного текста, совершаемые переводчиком в процессе перевода: такие отклонения фиксируются при сопоставлении текста перевода и оригинала и могут проявляться на разных уровнях языковой системы (Основные понятия переводоведения…, 2010, с. 126).

В.Н. Комиссаров, видный представитель лингвистической теории перевода, неоднократно затрагивает категорию ошибки в рамках рассмотрения возможностей анализа качества перевода. Хотя сам В.Н. Комиссаров не формулирует простого, узкоспециализированного определения «переводческой ошибки» в лексикографическом духе, из его научных трудов следует, что ошибка понимается как нарушение параметров адекватности и точности передачи содержания оригинала, включающее лексические и грамматические расхождения, а также искажения смысловых и функциональных характеристик текста (Комиссаров, 1990, с. 51–53).

Классическим представлением о переводческой ошибке в отечественной литературе о переводе можно считать подход Л.К. Латышева, который рассматривает перевод как производный текст, воспроизводящий содержательные и коммуникативные характеристики оригинала в условиях иной языковой системы. В рамках такого понимания перевода ошибка неизбежно возникает там, где система перевода не обеспечивает требуемой репрезентации исходного содержания, будь то на уровне лексики, грамматики или коммуникативной функции. Хотя сам Л. К. Латышев и не предлагает полного и четкого определения переводческой ошибки, его подход к толкованию переводческой деятельности создаёт базу для понимания ошибки как отклонения от требуемой технологической эффективности перевода (Латышев, 2005, с. 87).

Классик отечественного переводоведения Л.С. Бархударов в своей монографии «Язык и перевод» разрабатывает семантико-семиотическую модель перевода, в которой важное место отводится понятию адекватного смыслового соответствия между исходным текстом и переводом. В этой парадигме переводческая ошибка может быть охарактеризована как нарушение семантической и эквивалентной связи между элементами текста оригинала и их интерпретацией в переводе, а ошибки на уровне семантики ведут к искажению репрезентируемой информации (Бархударов, 1975, с. 68).

Исследователи Н.А. Матвеев и Г.А. Шушарина указывают на тот факт, что переводческие ошибки возникают в результате «когнитивных и лингвокультурных расхождений между языками, а также несовпадения речевых норм и картин мира» (Матвеев, Шушарина, 2015, с. 45). Это весьма важное наблюдение ученых подчеркивает, что переводческая ошибка – это результат сложного взаимодействия языковых, культурных и когнитивных факторов, а также возможного нарушения принципов эквивалентности, информационной точности и функциональной адекватности перевода.

А.В. Жидков в своей научной статье о переводе научно-технических текстов выделяет типичные ошибки, возникающие при передаче многокомпонентных терминов и аббревиатур, не имеющих прямых аналогов в целевом языке. Он отмечает, что «переводчик должен уметь адаптировать текст с учётом функционального назначения и целевой аудитории» (Жидков, 2014, с. 97).

В современной лингвистической традиции исследование ошибок входит в задачи отдельного раздела языкознания – эрратологии, понимаемой как общая теория ошибок. Существенно, что термин «эрратология» закрепляется и в переводоведческом употреблении. Он применяется для обозначения «теории ошибок и переводческих несоответствий» (Щелокова, 2020, с. 170), что позволяет рассматривать ошибку как объект систематического описания на уровне теории. При этом подчёркивается, что сам термин может использоваться и в широком, и в узком

понимании, причём более узкая трактовка связывает эрратологию прежде всего с анализом ошибок в передаче смысла и с задачами совершенствования языковой подготовки (там же).

В переводоведении эрратологический ракурс поддерживается и тем, что существует корпус исследований, ориентированных на классификацию переводческих ошибок. Так, приводится обзор подходов, при которых предлагаются две системы классификации переводческих ошибок: «статическая» (ошибки в морфологии, семантике и прагматике), соотносимая с переводческим процессом, и «динамическая» (неправильное понимание, неправильное превращение, неправильное выражение), соотносимая с результатом перевода (там же). В логике данного представления эрратологию интересует рамка, в которой ошибка описывается как структурируемое явление, допускающее разложение по уровням и по этапам переводческой деятельности.

Отметим, что эрратология встраивается и в дискурс оценки качества перевода. Так, в коллективной монографии под ред. Л.В. Кушниной глава «Эрратология перевода» (автор Л.П. Раскопина) включена в раздел, посвящённый освещению вопросов оценки качества перевода. Внутри этого подхода фиксируется, что ошибки в переводческой коммуникации могут быть обоснованными или необоснованными, случайными или закономерными, а их возникновение трактуется как естественный компонент языкового поведения участников перевода (Распокина, 2020, с. 257). Одновременно даётся максимально «широкая» дефиниция переводческой ошибки как необоснованного отступления от нормативного требования эквивалентности, т. е. как меры несоответствия перевода оригиналу и меры его дезинформирующего воздействия на адресата (там же). Тем самым эрратология задаёт нормативно-оценочную позицию, позволяющую описывать отклонения перевода через их эффект и через соотнесённость с требованием эквивалентности.

Вместе с тем, в самой эрратологической перспективе подчёркивается необходимость аккуратного выбора модели описания ошибки. В обсуждении подхода

А.Б. Шевнина к рассмотрению переводческих ошибок А.А. Яковлев фиксирует формулу, согласно которой переводческая ошибка «выражает определённый характер перехода от того, что задано “на входе” (в оригинале), к тому, что получено “на выходе” (в переводе)» (Яковлев, 2010, с. 111). Однако далее подчёркивается, что одно лишь сопоставление исходного текста и перевода не даёт достаточного основания для построения адекватной модели перевода (там же, с. 112).

Эрратология укрепляет теоретический аппарат описания «ошибки» как диагностируемого несоответствия, но одновременно подготавливает переход к тем категориям, которые позволяют описывать качество перевода шире, чем сумма локальных несоответствий (в частности – к «переводческой неудаче»).

Что касается зарубежной традиции, то понятие «translation error» (переводческая ошибка) также широко обсуждается в контексте теории оценки качества перевода. Так, в статье Г. Хансена термин *error* рассматривается как отклонение от требований и норм взаимодействия исходного и переводного текста, выражающееся в самых разных формах: от лексико-семантических и стилистических до прагматических и идиоматических. Такая типология подчеркивает, что ошибка может включать нарушения функции текста в коммуникативном контексте, что приближает зарубежные подходы к более широкому описанию переводческих ошибок, чем исключительно формальные критерии (Hansen, 2015).

Однако в ряде исследований используется и другой подход, при котором ключевым ориентиром перевода является сохранение смысла исходного высказывания (Sulistiyaningsih, 2025; Gili, 2023; Ngoc, 2021): содержание текста перевода должно соотноситься с содержанием оригинала, а переводчик не вправе произвольно изменять значение исходного текста. Вместе с тем перевод рассматривается как важнейший механизм межъязыковой и межкультурной коммуникации, поскольку именно он обеспечивает возможность передачи информации между носителями разных языков и культур и тем самым делает международное общение возможным.

Категория качества перевода в современной традиции выступает методологическим «узлом», поскольку именно с ней остальные переводоведческие категории (в том числе и понятие переводческой ошибки) соотносятся как с более общим основанием анализа. А.Ф. Шаповалова прямо фиксирует субординационный статус качества перевода. Обсуждение ошибок оказывается закономерным лишь тогда, когда задана рамка оценки качества и внятны принципы оценки качества перевода, а также принята типология переводческих ошибок как инструмент, позволяющий описывать достоинства и недостатки перевода в релевантной системе координат (Шаповалова, 2019, с. 305–306). В этом контексте ошибка выступает как элемент более широкой модели оценки качества, ориентированной на выявление степени соответствия перевода определённым требованиям.

С методологической точки зрения принципиально значимо разведение эквивалентности и адекватности как разных опорных категорий оценки. В изложении А. Ф. Шаповаловой эквивалентность понимается как результат сопоставления фрагментов оригинала и перевода и тем самым тяготеет к «сопоставительному» измерению качества, тогда как адекватность описывает соотнесённость перевода с условиями конкретной коммуникативной ситуации и задаёт «ситуативно-функциональный» критерий (Шаповалова, 2019, с. 306). Эта логика вполне согласуется и с положением отечественного классика перевода А. Д. Швейцера о том, что для теории перевода существенны обе категории: эквивалентность фиксирует отношение двух текстов, а адекватность соотносит перевод с условиями коммуникации; при этом адекватность прямо связывается со стратегическим выбором и определяется «стратегией перевода» (Швейцер, 1988, с. 94). В аксиологическом ракурсе такой сдвиг дополнительно поддерживается тем, что в ряде современных подходов эквивалентность уже не рассматривается как «базовое» понятие, а вместо неё в фокус выводится адекватность как более функциональный ориентир качества (Соколова, 2020).

Отсюда следует, что оценка качества не может быть «унифицированной» процедурой: она соотносится с видом перевода, типом текста и с тем, какие параметры считаются релевантными в данной сфере. Так, А.Ф. Шаповалова показывает, что для профессионального перевода критериями оценки качества перевода могут выступать (а) доля ошибочного перевода, (б) успешность осуществления предметной деятельности коммуникантов и инициатора перевода и (в) удовлетворение потребностей/ожиданий адресата (клиента, получателя, потребителя) (Шаповалова, 2019, с. 308).

Наряду с теоретическими моделями оценка качества перевода имеет и институциональное измерение, отражённое в стандартах и процедурах контроля качества. В частности, в учебнике П. П. Дашинимаевой «Теория перевода. Психолингвистический подход» отмечается, что европейский стандарт EN 15038:2006 регламентирует требования к переводческой деятельности, включая профессиональные компетенции переводчика, организацию переводческого процесса и контроль качества результата (Дашинимаева, 2017, с. 241–242).

В зарубежной традиции Translation Quality Assessment (оценка качества перевода) вопрос качества формулируется как вопрос о критериях «хорошести» перевода, что приводит к существованию различных моделей оценки – от интуитивно-интерпретативных и реактивных (response-based) до дискурсивных и нормативно-описательных (House, 2001, p. 243–244). В совокупности эти подходы демонстрируют, что качество перевода не сводится к отсутствию ошибок, а определяется тем, насколько перевод отвечает требованиям выбранной модели оценки и обеспечивает выполнение своей функции в конкретной коммуникативной ситуации.

Таким образом, в традиции переводоведения переводческая ошибка представляется как структурно-описательная категория, с помощью которой фиксируются отклонения от параметров точности, эквивалентности и функционального соответствия между исходным текстом и переводом. Она

охватывает разнообразные уровни языковых и смысловых отношений и применяется в качестве диагностического инструмента при анализе качества переводных текстов в сопоставительной перспективе. Общая характеристика переводческой ошибки в классических источниках связывается с несоответствием между ожидаемым результатом перевода и фактически полученным текстом, которое измеряется благодаря категориям эквивалентности, адекватности и точности.

1.5.2. Классификации переводческих ошибок: обзор

В переводоведении классификации переводческих ошибок традиционно строятся вокруг разных оснований: одни ученые исходят из степени и характера смыслового расхождения между оригиналом и переводом, другие – из уровня языковой организации, третьи – из функционально-коммуникативной результативности перевода в конкретной ситуации. Именно множественность оснований может объяснить, почему в научной литературе сосуществуют разнообразные типологии: они фиксируют разные аспекты переводческого результата и по-разному соотносят «ошибку» с качеством перевода и мерой её влияния на адресата.

В рамках классической лингвистической теории перевода распространён подход, при котором переводческие ошибки ранжируются по характеру нарушения эквивалентности и по последствиям этого для понимания смысла. Так, в обзоре классификации ошибок, сделанном Л. П. Раскопиной, переводческая ошибка в самом общем виде описывается как «необоснованное отступление от нормативного требования эквивалентности», а также как «мера несоответствия перевода

оригиналу» и «мера дезинформирующего воздействия на читателя» (Раскопина, 2020, с. 116). Такая постановка вопроса важна методологически: она задаёт связь между классификацией ошибок и оценкой качества перевода через призму «эффекта для адресата» и «степени искажения» исходного содержания.

Наиболее известная и часто воспроизводимая в учебной и научной традиции классификация ошибок по степени смыслового расхождения фиксирует три базовые группы: искажения (distortions), неточности (inaccuracies) и неясности/двусмысленности (ambiguities). В научном анализе переводческих ошибок В. К. Терентьевой и Е. Е. Васильева этот принцип описывается как классический для отечественной школы: ошибки различаются по тому, приводят ли они к подмене смысла, к частичной потере/смещению смыслового компонента или к размыванию/неопределённости сообщения в переводе (Терентьева, Васильев, 2020, с. 56–57]. Достоинство данной схемы, несомненно. Прослеживается её тесная связь с семантическим результатом и возможностью «взвешивания» ошибок по тяжести. Однако возникающее при этом ограничение проявляется прежде всего в том, что представляемая схема учитывает преимущественно содержательный итог и в меньшей степени прослеживает случаи, когда перевод формально не искажает денотативный смысл, но изменяет прагматический вектор, жанровую тональность или адресатную управляемость текста (т.е. параметры, критичные для прикладных аспектов перевода).

Другая группа классификаций переводческих ошибок строится в зависимости от уровней языка и «локализации» ошибок. Разграничиваются лексические, грамматические, стилистические и др. типы ошибок. Ю.Е. Выволокина, обобщая различные подходы к классификации переводческих ошибок, показывает, что в литературе переводческая ошибка нередко трактуется именно как отклонение, фиксируемое на конкретных языковых уровнях, причём само понятие ошибки соотносится и с нормой, и с узусом, и с требованиями эквивалентности (Выволокина, 2020, с. 1–2). Такой «уровневый» принцип удобен для систематизации

материала и сопоставления типичных нарушений, но при его применении возникает методологически важный момент: локализация ошибки на уровне формы ещё не гарантирует адекватного объяснения того, почему данное решение оказалось неудачным и каков его коммуникативный эффект в употреблении текста перевода.

Отметим, что в рамках современных исследований последнего времени усиливается стремление описывать ошибки как сбои переводческого замысла и рассогласование с целевой функцией. Так, И.А. Баринова и И.Г. Овчинникова на материале анализа переводческой деятельности предлагают разделять ошибки трёх типов: языковые, когнитивные и коммуникативные, подчёркивая при этом важную и принципиальную, на наш взгляд, мысль: «коммуникативные ошибки отражают сбой в стратегии перевода», тогда как когнитивные и языковые соотносятся скорее с тактикой (Баринова, Овчинникова, 2015, с. 94). Авторы показывают, что «формально верный перевод без языковых и предметных ошибок» способен не выполнять исходную коммуникативную функцию в принимающей культуре, т. е. обнаруживает дефект не на уровне «знака», а на уровне дискурсивной результативности. Тем самым вводится иной критерий типологизации переводческих ошибок: ошибка распознаётся через то, занимает ли перевод «предназначенную ему нишу» в целевом коммуникативном пространстве (там же).

Сходное смещение фокуса – от «ошибки как расхождения» к «ошибке как маркеру некорректного стратегического курса» – демонстрируется и в статье «От переводческой ошибки к переводческой стратегии». В ней, опираясь на подход Д. М. Бузаджи, В. В. Гусева, В. К. Ланчикова и Д. В. Псурцева, авторы воспроизводят типологию ошибок, в которой они распределяются по четырём группам: искажения фактов, искажения логики, искажения стилистики и искажения языковой нормы (Руденко, Темирханова, Исаева, 2021, с. 130). Данная классификация комбинирует содержательные и формально-языковые критерии, а также явно включает логический компонент как самостоятельную область нарушений. Подобная типология переводческих ошибок позволяет показать, что переводческая ошибка

может быть описана как нарушение когнитивно-логической связности, что напрямую связано с управлением переводческими решениями.

Вместе с тем, сравнительный анализ классических и «комбинированных» классификаций выявляет несколько методологически значимых расхождений. Во-первых, разграничивается то, что признаётся объектом оценки: либо соотношение текстов «оригинал – перевод», либо перевод как инструмент функционирования в ситуации. Во-вторых, различается логика «веса» ошибки: либо вес выводится из степени смыслового искажения (искажение/неточность/неясность), либо на основании того, приводит ли решение к сбою в реализации цели и жанровой модели. В-третьих, неодинаково представлена область «прагматики»: в одних классификациях она является частью стилистики/эквивалентности, в других – выделена как самостоятельное измерение.

Эти расхождения становятся особенно заметны при переносе существующих классификаций ошибок на тексты, ориентированные на практическое действие адресата. В документации по веб-разработке переводной текст функционирует как средство организации действий, обучения и корректной интерпретации процедур. Как уже было отмечено ранее, часть переводческих решений может быть формально корректной и при этом приводить к снижению коммуникативной эффективности, изменению тональности, утрате обучающей направленности или нарушению жанровых ожиданий адресата. Эти наблюдения важны в контексте границ применимости традиционных классификаций ошибок. Схемы, ориентированные преимущественно на «смысловую точность» и «языковую правильность», оказываются недостаточно чувствительными к тем отклонениям, которые проявляются как функциональная несостоятельность текста в профессиональной среде пользователя.

В отношении документации по веб-разработке принципиальным становится расширение классификационного основания: помимо семантических расхождений и уровневых нарушений необходимо фиксировать отклонения, которые затрагивают

коммуникативную задачу текста, его прагматический дизайн. В этом плане релевантно сопоставить классический подход «по степени смыслового искажения» с коммуникативно-стратегическим: если в первом случае ошибка мыслится как несоответствие требованиям эквивалентности, то во втором – как проявление сбоя стратегического согласования переводческих решений, влияющего на способность текста занять корректную функцию в целевом дискурсе.

В более современных зарубежных исследованиях понятие переводческой ошибки осмысляется уже как результат действия более широких механизмов переводческого процесса. Так, Ц. Сюй предлагает рассматривать переводческие ошибки через рамку субъективных и объективных факторов. К субъективным факторам автор относит физиологические ограничения переводчика, включая утомление и колебания работоспособности, психологические состояния, такие как эмоциональная нестабильность и ослабление внимания, культурно обусловленные особенности восприятия, а также когнитивные механизмы, определяющие эффективность двуязычной обработки информации. В свою очередь, к объективным факторам Ц. Сюй относит языковую сложность исходного текста, в том числе терминологическую неоднозначность и культурно-специфические элементы, технологические ограничения переводческих инструментов, а также особенности рабочей среды, включая отвлекающие факторы и неудобства организационного характера. Тем самым обсуждение переводческих ошибок выводится за пределы сугубо нормативного понимания ошибки и связывается с его многофакторной природой. Вместе с тем даже при таком расширении аналитической рамки центральной категорией у исследователя остаётся именно «ошибка» (error), то есть отклонение, подлежащее выявлению и объяснению в пределах модели ошибки (Xu, 2025). Между тем в текстах документации по веб-разработке существенная часть переводческих отклонений проявляется скорее не как однозначно фиксируемая ошибка, а как постепенное смещение функциональной результативности текста. Это делает необходимым обращение к более широкой категории переводческой неудачи,

которая позволяет учитывать последствия для функционирования перевода в профессиональной среде пользователя.

Вместе с тем представляется важным учитывать, что расширение рамки анализа за счёт субъективных факторов фактически выводит проблему переводческих отклонений в психологическое измерение. Если ошибка в традиционном понимании обычно фиксируется как результат несоответствия между оригиналом и переводом, то переводческая неудача позволяет увидеть за этим результатом действие когнитивных и эмоциональных механизмов переводческой деятельности. В этом случае в поле анализа попадают особенности внимания, утомление, напряжение, колебания работоспособности, специфика восприятия и интерпретации материала, то есть те параметры, которые не всегда приводят к явной ошибке, но могут постепенно снижать функциональную состоятельность перевода. Тем самым категория переводческой неудачи оказывается методологически значимой ещё и потому, что позволяет рассматривать перевод как деятельность, в которой языковые, коммуникативные и психологические факторы тесно взаимосвязаны.

1.5.3. Понятие переводческой неудачи и его методологический потенциал

В ряде современных исследований переводческая неудача трактуется как сбой в реализации коммуникативного и прагматического потенциала перевода, возникающий вследствие рассогласования параметров текста оригинала и текста перевода. Так, Л. К. Латышев и Н. Ю. Северова подчёркивают, что даже при формальном соблюдении языковой нормы «нарушение узуса приводит к коммуникативным неудачам», поскольку адресат не получает эквивалентного

коммуникативного эффекта (Латышев, Северова, 2017, с. 297). В этом случае перевод может оставаться грамматически и лексически корректным, однако не выполняет своей социальной и прагматической функции, что позволяет говорить именно о переводческой неудаче, а не об ошибке в традиционном нормативном понимании.

Аналогичная позиция относительно понимания переводческой неудачи развивается в научной статье Т. М. Софроновой и Т. П. Бабак, в которой подчёркивается, что недостаточная развитость прагматической компетенции переводчика приводит к так называемой «прагматической неудаче», возникающей даже при отсутствии явных лексических или грамматических искажений (Софронова, Бабак, 2022, с. 16). Авторы связывают такие неудачи с несоответствием перевода коммуникативному намерению оригинала и ожиданиям адресата, подчёркивая приоритет прагматической нормы в иерархии переводческих норм.

Подобное понимание переводческой неудачи позволяет сместить фокус анализа с субъективной «вины» переводчика на системные факторы, включая жанровые конвенции, условия коммуникации и специфику предметной области. В ряде исследований цифрового перевода показано, что коммуникативные неудачи возникают «на всех языковых уровнях» и обусловлены в том числе социокультурными причинами, а также особенностями медиума и языков-посредников (Карпенко, 2025, с. 269; Полякова, 2012, с. 89; Лифэнь, 2025). В результате переводческая неудача видится как аналитическая категория, позволяющая описывать отклонения перевода от коммуникативной адекватности без сведения их к отдельным нарушениям норм, а значит к языковым ошибкам на межязыковом уровне и без апелляции к индивидуальной несостоятельности переводчика. Иными словами, неудача не тождественна ошибке и не сводится к ней, поскольку может иметь место даже в тех случаях, когда перевод формально корректен и не содержит очевидных лексических или грамматических отклонений.

Методологическое обоснование использования термина «переводческая неудача» в значительной степени связано с использованием в нашем исследовании

идей интегративного переводоведения, в рамках которого перевод рассматривается как многоаспектное явление, требующее одновременного учета разных групп факторов, влияющих на переводческую деятельность. Так, А. М. Поликарпов напрямую связывает развитие интегративного переводоведения с новым видением переводческой деятельности как «динамической целенаправленной системы», внутри которой «влияющие на перевод внутренние и внешние факторы комплексно изучаются с учетом творческой личности переводчика, многослойности его мышления, а также текстоцентрического принципа и комплексной оценки качества перевода» (Поликарпов, 2017, с. 7). Важной методологической опорой интегративного подхода становится и принцип «многоканальности» мышления переводчика, который «предполага[ет] интеграцию сознательных ментальных действий с бессознательными операциями», что позволяет характеризовать переводческую деятельность «как целенаправленную динамическую систему взаимосвязанных действий» (там же). Следует отметить, что при такой постановке вопроса фокус переводческого исследования смещается, действительно, на качество переводческого результата. Не случайно А.М. Поликарпов подчеркивает «ориентаци[ю] как в исследованиях, так и в преподавании на качество переводческой деятельности с учетом многих факторов, влияющих на нее» (там же). В интегративной научной парадигме переводческая неудача может быть интерпретирована как удобная единица аналитического уровня, поскольку она позволяет фиксировать рассогласования в переводе как «динамической системе» и обсуждать качество перевода в логике комплексного учета факторов, на которые ориентируются представители интегративного переводоведения.

Существенным для понимания методологического потенциала понятия неудачи является его соотнесённость с теорией норм и описательным переводоведением. В научных трудах Г. Тури нормы перевода трактуются как социально обусловленные ожидания и регулярности переводческого поведения, действующие в определённой культуре и жанровой традиции (Toury, 1995, с. 54–56). В этом контексте

переводческая неудача может быть интерпретирована как несоответствие перевода не столько формальным языковым правилам, сколько нормам функционирования текста в принимающей культуре. Такой подход позволяет рассматривать отклонения перевода в динамике профессионального и культурного контекста, а не исключительно в координатах «верно – неверно», «правильно – неправильно».

Важно подчеркнуть, что особую методологическую значимость понятие переводческой неудачи приобретает при анализе текстов перевода, для которых функциональная направленность имеет первостепенное значение. К числу таких текстов относится документация по веб-разработке, которая одновременно выполняет справочную и обучающую функции и ориентируется на предполагаемые действия адресата. В этом отношении показательна и сама логика учебного перевода, цель которого заключается в формировании у студентов переводческих компетенций, необходимых для их будущей профессиональной деятельности (Сдобников, 2015, с. 87–88). Подобная установка ещё раз подтверждает, что перевод должен оцениваться в первую очередь с точки зрения функциональной пригодности текста в конкретной сфере применения. В текстах документации по веб-разработке важны тональность изложения, степень директивности, прозрачность инструкций и соответствие используемому профессиональному регистру. Формально допустимый перевод, сохраняющий в основном передаваемую информацию исходного текста, может при этом изменять прагматический вектор интенции текста оригинала, ослаблять его инструктивный характер или нарушать жанровые ожидания профессионального сообщества, что приводит к снижению эффективности трансфера знаний.

Принципиально важное развитие этих идей представлено в коммуникативно-функциональном подходе, последовательно разрабатываемом В. В. Сдобниковым, в рамках которого категория качества перевода рассматривается как центральная и методологически определяющая. В.В. Сдобников подчёркивает, что качество перевода не может оцениваться вне конкретной коммуникативной ситуации, поскольку «перевод осуществляется в разных условиях, в разных коммуникативных

ситуациях, и уже поэтому перевод конкретного текста не может быть хорошим или плохим “всегда и везде”» (Сдобников, 2015, с. 41). Исходя из этого, ученый формулирует универсальное требование к качеству перевода – его способность удовлетворять потребности коммуникантов и инициатора перевода в рамках их предметной деятельности в определённой коммуникативной ситуации (там же). Подобное понимание качества перевода позволяет трактовать переводческую деятельность как процесс принятия решений в условиях неопределённости, когда переводчик соотносит свои действия с предполагаемыми ожиданиями адресата и условиями использования перевода.

Существенным для анализа переводческих отклонений является справедливо осуществляемая В.В. Сдобниковым критика текстоцентрического подхода к оценке качества перевода, при котором перевод сопоставляется с оригиналом как автономный текст, рассматриваемый в отрыве от реальной коммуникативной ситуации. По мнению автора, такой подход не позволяет определить «весомость» ошибки так, как это происходит в профессиональной практике, поскольку не учитывает цели перевода и условия его функционирования (там же, с. 42). В результате отклонение фиксируется формально, однако его реальное влияние на успешность коммуникации и предметной деятельности адресата остаётся вне поля анализа. В коммуникативно-функциональной модели перевода оценка его качества ориентирована на полезность перевода для осуществления деятельности адресата, что позволяет интерпретировать переводческую неудачу как случай, при котором перевод, оставаясь формально допустимым, оказывается нефункциональным или недостаточно эффективным в заданной коммуникативной среде. Вместе с тем в отношении документации по веб-разработке данная позиция представляется особенно убедительной при условии учёта внутренней организации текста, поскольку коммуникативная ситуация и текстовая форма в подобных материалах тесно соотнесены.

1.5.4. Переводческое стратегирование

В результате проведённого разграничения переводческих ошибок и переводческих неудач становится очевидным, что многие отклонения в переводе, не сводимые к формальным нарушениям языковой нормы, требуют объяснения на ином уровне анализа. В этих случаях речь идёт не столько о некорректности отдельных переводческих решений, сколько о несоответствии общего направления перевода условиям конкретной коммуникативной ситуации, целям перевода и ожиданиям адресата. Подобное понимание переводческих отклонений согласуется с функционально ориентированными подходами к оценке качества перевода, в рамках которых отклонение интерпретируется через результат переводческой деятельности, а не через отдельные языковые несоответствия. Именно такая логика позволяет рассматривать переводческие неудачи как системные явления, затрагивающие разные аспекты текста перевода.

Весьма важным в данном контексте представляется обращение А. М. Поликарпова к понятию переводческого стратегирования, позволяющему описывать перевод как процесс целенаправленного выбора, согласования и последующей коррекции переводческих решений. Понятие переводческого стратегирования последовательно разрабатывается А.М. Поликарповым в рамках интегративного переводоведения и соотносится с рассмотрением перевода как деятельности, в которой взаимодействуют внутренние параметры переводческого процесса и внешние условия коммуникации. В частности, А.М. Поликарпов подчёркивает, что интегративное переводоведение позволяет объяснять процесс перевода через «взаимную активизацию ментальных и

вербальных единиц» во время переводческого стратегирования, включающего, среди прочего, реализацию переводческих макро- и микростратегий с учётом ситуационных факторов на этапах подготовки к переводу, собственно перевода и саморедактирования (Поликарпов, 2025, с. 218). Тем самым стратегия перестаёт пониматься как статичный набор приёмов: она осмысляется как способ реализации более широкого процесса стратегирования, организующего деятельность переводчика и обеспечивающего связность переводческих действий.

Принципиально важным положением концепции А.М. Поликарпова является разграничение переводческого стратегирования и переводческой стратегии. А.М. Поликарпов фиксирует сам предмет обсуждения как систематизацию знаний о сознательных и бессознательных компонентах переводческого стратегирования и подчёркивает, что переводчик действует в режиме многозадачности и многоканальности мышления (там же). Это означает, что стратегическое направление перевода далеко не всегда формулируется эксплицитно, но при этом переводческий процесс опирается на определённые когнитивные «установки, из которых сознательно или бессознательно исходит переводчик», и которые задают основу его переводческого поведения в конкретных условиях (там же). При таком понимании стратегирование становится удобным объяснением для тех случаев, когда наблюдаемое отклонение проявляется как результат более общего рассогласования в организации переводческой деятельности.

Содержательное наполнение понятия переводческого стратегирования А.М. Поликарпов раскрывает, опираясь на критический анализ предшествующих подходов к стратегии перевода и к стратегическому компоненту деятельности переводчика. Автор показывает, что уже в зарубежной традиции переводческая стратегия осмыслялась как совокупность потенциально осознаваемых планов переводчика, ориентированных на решение переводческой задачи; при этом различались макростратегия как линия поведения, охватывающая несколько задач, и микростратегия как способ решения частной задачи (там же, с. 220). Существенно

также и то, что в отечественной традиции стратегия связывается с наличием «генеральной линии поведения», основанной на осознании конечной цели перевода, а выбор стратегии – с такими факторами, как цель перевода, социальная норма перевода и жанр переводимого текста (там же). А.М. Поликарпов правомерно обращает внимание на коммуникативно-функциональную трактовку стратегии перевода как программы осуществления переводческой деятельности, формирующейся в зависимости от коммуникативной ситуации и цели перевода; при этом в качестве компонентов стратегии называются ориентирование в ситуации, формулирование цели, прогнозирование возможных изменений ситуации и реакций участников коммуникации, планирование и др. (там же). Тем самым «стратегическое» измерение перевода задаётся целостной программой действий, выстраиваемой относительно цели, адресата и условий функционирования перевода.

Особое значение для понимания механизма переводческого стратегирования имеет то, что А.М. Поликарпов прямо связывает его с сочетанием рефлексивных и рефлекторных (то есть не полностью осознаваемых) компонентов переводческого поведения и подчёркивает необходимость учитывать одновременно и алгоритмизацию последовательности действий, и участие эвристической составляющей в разработке и реализации стратегии (там же). Это положение позволяет концептуализировать стратегирование как процесс, в котором формально-логические, «алгоритмизированные» операции не существуют отдельно от интуитивных и поисково-эвристических, а напротив, образуют единый механизм принятия решений, а их соотношение определяется ситуацией перевода и профессиональным опытом переводчика.

Ключевой методологической опорой дальнейшего рассуждения у А.М. Поликарпова становится собственно дефиниция переводческого стратегирования. Отталкиваясь от общенаучного понимания стратегирования как множества одновременных и последовательных действий на разных уровнях и подчёркивая важность его интерпретации как ансамбля стратегического мышления,

действия и получения знаний, А.М. Поликарпов вводит рабочее определение переводческого стратегирования. Под переводческим стратегированием предлагается понимать «осознанное или бессознательное ментальное маневрирование переводчика, выстраивание стратегий перевода в зависимости от цели перевода, ситуации, условий и внутренних и внешних факторов, с гибким ориентированием на генеральную линию, планированием переводческой деятельности с возможностью прогнозирования и управления, а также с возможностью корректировки и модификации» (там же, с. 222). В данной формулировке существенны сразу несколько моментов: стратегирование включает ориентацию на цель и ситуацию, предполагает удержание «генеральной линии», включает прогнозирование и управление, а также допускает коррекцию и модификацию как имманентный компонент переводческой деятельности.

Понятие переводческого стратегирования соотносится с представлением о переводческом мышлении как важном компоненте переводческой деятельности. А. М. Поликарпов рассматривает переводческое мышление с позиций интегративного переводоведения и связывает его с работой переводческой личности, направленной на понимание исходного текста, определение цели перевода, выбор переводческих решений и оценку их соответствия условиям коммуникации (Поликарпов, 2026, с. 131–135). В связи с этим переводческая неудача может рассматриваться как результат недостаточной согласованности переводческого решения с коммуникативной задачей текста, особенностями адресата, жанровой природой документации и функциональным назначением перевода.

Дальнейшее развитие концепции в теории переводческого стратегирования позволяет ее автору уточнить когнитивную природу стратегирования. А.М. Поликарпов отмечает, что многие переводческие решения обычно реализуются на стыке алгоритмизированных формально-логических и поисково-эвристических, интуитивных операций; при этом алгоритмизированное (осознаваемое) и эвристическое (неосознаваемое) начала тесно переплетаются (Поликарпов, 2025).

Существенно также, что набор стратегий, входящих в переводческое стратегирование, может рассматриваться как индивидуальный, а «список возможных стратегий перевода» – как открытый; при этом стратегирование во многом зависит от коммуникативной ситуации, временной и пространственной ориентации переводчика и семиотического характера коммуникации, что обусловливает проявление как перспективных, так и ситуативных стратегий (там же). Тем самым стратегирование описывается как гибкая система действий, постоянно соотносимая с параметрами ситуации и допускающая вариативность, а не как фиксированный «набор правил».

Логическим продолжением этой линии является вывод о необходимости постоянного вариативного поиска в процессе перевода и о роли накопленного опыта в управлении стратегическим курсом. А.М. Поликарпов подчёркивает, что умение соотносить поступающую информацию с накопленным опытом помогает переводчику «лавировать», придерживаясь генеральной линии, и избегать «подводных камней» (там же, с. 223). В качестве основания переводческого стратегирования описывается комплекс когнитивных механизмов, включающих осознанную рефлексию, креативность и переводческую интуицию, что в совокупности формирует базу для стратегирования как системы, включающей когнитивные установки, стратегии перевода и ментальные операции и приводящей к выработке переводческих решений (там же). Далее А.М. Поликарпов делает вывод о том, что введение понятия переводческого стратегирования в научный оборот интегративного переводоведения закономерно, поскольку оно позволяет описывать способность переводчика ориентироваться в ситуации, распознавать паттерны и демонстрировать навыки принятия переводческих решений (там же). При этом подчёркивается, что стратегирование осуществляется при непрерывном взаимодействии сознательных и бессознательных компонентов, позволяя переводчику учитывать множество факторов переводческой ситуации, осуществлять контроль за своими действиями и при необходимости корректировать вербальное

поведение; в случае письменного перевода отдельно отмечается роль саморедактирования как элемента обеспечения качества перевода (там же).

В контексте анализа переводческих неудач такая концепция переводческого стратегирования обладает высокой объяснительной ценностью. Поскольку стратегирование включает удержание генеральной линии, ситуативную адаптацию и постоянную корреляцию действий с целью и условиями коммуникации, переводческая неудача может быть интерпретирована как следствие рассогласования между стратегической установкой и реальными параметрами функционирования перевода либо как результат недостаточной согласованности действий переводчика внутри процесса стратегирования. В этом случае наблюдаемое отклонение выражается не обязательно в виде отдельных ошибок, но в снижении функциональной состоятельности перевода и его коммуникативной эффективности в конкретной сфере употребления, что особенно существенно для текстов прикладного характера.

Применительно к переводу документации по веб-разработке обращение к понятию переводческого стратегирования представляется методологически оправданным. Документация как гибридная разновидность текста требует согласования обучающей и справочной функций, точности терминологии и доступности изложения, а также ориентации на профессиональное действие адресата. Несогласованное или непоследовательное стратегирование в переводе таких текстов приводит к тому, что формально корректные переводческие решения оказываются функционально неэффективными и препятствуют полноценному трансферу знаний. В таком случае переводческие неудачи целесообразно рассматривать как проявление стратегических сбоев, что создаёт теоретическую основу для последующего типологического анализа переводов документации по веб-разработке.

1.5.5. Типология переводческих неудач

Логическим продолжением рассмотрения переводческого стратегирования как процесса целенаправленного (в том числе частично неосознаваемого) управления переводческими решениями становится необходимость описать те типы отклонений, которые проявляются, прежде всего, как сбои в реализации функционально значимого результата перевода. В коммуникативно-функциональной модели переводческого процесса качество перевода принципиально связывается с условиями конкретной коммуникативной ситуации и с потребностями участников деятельности, вследствие чего перевод не может оцениваться как «хороший» или «плохой» вне заданных условий употребления (Сдобников, 2015). При этом сама процедура фиксации отклонений, ориентированная только на сопоставление двух текстов, ограничена, поскольку не позволяет в достаточной степени учитывать «весомость» отклонения в реальной практике и его влияние на успешность предметной деятельности адресата. В работах, посвящённых методологии оценки качества перевода, неоднократно подчёркивается, что категория качества является базовой, а остальные переводоведческие категории (включая ошибку) соотносятся с ней как с надстроечной рамкой (Сдобников, 2015; Княжева, 2010; Ramos, 2014). Принципиально важно также разведение эквивалентности и адекватности: если эквивалентность тяготеет к сопоставительному измерению, то адекватность определяется соотнесённостью перевода с условиями коммуникативной ситуации и может напрямую связываться со стратегическим выбором переводчика. На этом основании становится возможным переход от описания «переводческих ошибок» как преимущественно локальных несоответствий к описанию «переводческих неудач»

как отклонений, обнаруживаемых на уровне функциональной состоятельности текста перевода и его коммуникативного эффекта.

Предлагаемая автором настоящего исследования типология переводческих неудач ориентирована на специфику документации по веб-разработке как функционально направленного дискурса, в котором перевод выступает инструментом трансфера специализированного знания и организации действий адресата и соотносится с ранее предложенной типологией переводческих неудач в англоязычной документации по веб-разработке, в рамках которой учитываются лексико-семантические, прагматические, стилистические, синтаксические и гибридные неудачи, влияющие на качество перевода и функциональность текста документации по веб-разработке (Поликарпов, Вовченко, 2025). В подобных текстах существенными оказываются тональность, директивность, прозрачность инструкций и соответствие профессиональному регистру; формально допустимое решение может при этом менять прагматический вектор и снижать эффективность трансфера знаний.

Рассмотрим далее основные типы переводческих неудач, отталкиваясь от наиболее явно прослеживаемых к более абстрактным случаям, требующим более детальной интерпретации.

1.5.5.1. Лексико-семантические переводческие неудачи

Лексико-семантические неудачи охватывают такие отклонения в переводе, при которых то или иное переводческое решение приводит к смещению или размыванию специализированного содержания, нарушает устойчивые соответствия в

терминологической системе или формирует некорректные смысловые связи, препятствующие точному усвоению и применению знания. Их принципиальная особенность состоит в том, что они проявляются не обязательно как грубое «искажение фактов», а как систематическое снижение информативной и обучающей эффективности текста вследствие неудачного выбора номинаций, несовпадения объёма значения, смешения терминологического и общеязыкового употребления, а также некорректного распределения семантических акцентов. В рамках подходов, связывающих качество перевода с параметрами деятельности адресата, подобные решения значимы прежде всего тем, что они снижают полезность перевода как инструмента предметной деятельности, т.е. затрагивают функциональный критерий качества, а не только формальную сопоставимость. При ориентации на адекватность как ситуативно-функциональный критерий оценки качества лексико-семантическую неудачу можно описать как отклонение, затрагивающее способность текста перевода обеспечивать корректную интерпретацию профессиональных понятий и процедур в условиях конкретной коммуникации.

Подчеркнём, что выделение данного типа особенно оправдано применительно к документации по веб-разработке, поскольку здесь терминологическая точность и предсказуемость лексических номинаций непосредственно связаны с успешностью выполнения действий адресатом, а также с воспроизводимостью описываемых процедур. В отличие от художественного перевода, в котором вариативность лексической номинации может быть стилистически мотивирована, в документации лексическая вариативность часто требует отдельного функционального обоснования, иначе она превращается в источник ненужной когнитивной нагрузки и приводит к распаду «генеральной линии» объяснения. С точки зрения стратегирования лексико-семантические неудачи коррелируют с нарушением принципа удержания общей линии решений и с недостаточной корреляцией выбора единиц с целью перевода и условиями его использования. Иными словами, это показатели рассогласования

между стратегической установкой на точный трансфер знания и реальным способом реализации этой установки в тексте.

1.5.5.2. Прагматические переводческие неудачи

В рамках семиотической традиции прагматика обычно понимается как аспект знаковой деятельности, связанный с отношением знаков к их интерпретатору и к условиям их употребления. В классической модели Ч. Морриса прагматика противопоставляется синтактике и семантике и описывает не внутреннюю организацию знаковой системы и не значение знаков как таковое, а их функционирование в реальной коммуникации, то есть то, каким образом знак воздействует на адресата, в каких условиях используется и какие последствия вызывает его употребление (Morris, 1971; Niu, 2022). Будучи первоначально осмысленной в семиотике как аспект знаковой деятельности, связанный с отношением знаков к их интерпретатору, прагматика в дальнейшем получила развитие в лингвистике и теории речевой деятельности и стала пониматься как сфера, охватывающая коммуникативное намерение говорящего, установку на адресата, контекст употребления высказывания, ожидаемый эффект и параметры речевого воздействия (Поликарпов, 2014). Применительно к переводу это означает, что прагматика перевода связана с воспроизведением или функционально мотивированной модификацией тех коммуникативных параметров, которые определяют, как переводной текст будет воспринят, понят и использован в принимающей среде.

В переводоведении прагматический аспект особенно значим в тех случаях, когда переводной текст включён в практическую деятельность адресата и должен направлять действия, предупреждать о возможных ограничениях, задавать сценарии интерпретации и обеспечивать предсказуемый результат коммуникации. С этой точки зрения прагматика перевода заключается в соотнесении переводческих решений с коммуникативным намерением оригинала, характеристиками адресата, ситуацией употребления текста и тем функциональным эффектом, который должен быть достигнут в целевой среде (Сдобников, 2015). В интегративном переводоведении такая постановка вопроса представляется особенно значимой, поскольку перевод рассматривается как сложная динамическая система, зависящая от внутренних и внешних факторов, а оценка его качества предполагает учёт прагматических, когнитивных, дискурсивных и деятельностных параметров. При этом существенное значение приобретает и переводческое мышление, поскольку именно оно обеспечивает соотнесение языковой формы, коммуникативной задачи, предполагаемой реакции адресата и условий функционирования текста в профессиональной среде (Поликарпов, 2025).

Прагматические неудачи в этой связи целесообразно интерпретировать как отклонения, связанные с несоответствием перевода коммуникативному намерению оригинала, ожиданиям адресата и условиям употребления текста в целевой профессиональной среде. На данном уровне интерпретации ключевыми становятся параметры, выходящие за рамки локальных соответствий: степень директивности и инструктивности текста, распределение ответственности между автором и адресатом, способы маркирования ограничений и предостережений, моделирование типичных сценариев использования и другие компоненты прагматической организации текста. Прагматическая неудача принципиально возможна и без очевидных лексических или грамматических искажений, поскольку затрагивает не столько то, что сказано, сколько то, для чего это сказано и каким образом это должно сработать в реальной коммуникации.

Особую значимость такой подход приобретает при анализе документации по веб-разработке, поскольку коммуникация в ней ориентирована на предсказуемое действие пользователя и на предупреждение неверных интерпретаций, способных привести к ошибочным практикам. В текстах данного типа прагматические параметры тесно связаны с критерием адекватности, если понимать её как соотнесённость перевода с условиями конкретной коммуникативной ситуации и, следовательно, как функциональный критерий качества (Смирнова, 2017). В стратегическом измерении прагматические неудачи могут рассматриваться как индикаторы некорректной корреляции переводческих решений с целью и условиями коммуникации, то есть как проявления сбоев в стратегировании, включающем прогнозирование, управление и готовность к модификации линии решений (Поликарпов, 2025). Следовательно, прагматическая неудача описывает ситуацию, при которой перевод, оставаясь формально допустимым, функционирует хуже ожидаемого как инструмент деятельности. В этом смысле она представляет собой маркер несогласованности стратегического курса перевода с коммуникативным устройством жанра, с практиками профессионального сообщества и с логикой использования текста в цифровой профессиональной среде.

1.5.5.3. Стилистические переводческие неудачи

Исследование стилистических неудач в настоящей работе проводится в русле стилистики дискурса. Такое решение опирается на положение о том, что стилистический и дискурсивный подходы имеют общий предмет исследования (т.е. текст) и общий коммуникативный вектор его изучения, при этом «не исключают и не

дублируют друг друга, а взаимодополняют друг друга» (Клушина, 2016, с. 78). Для адекватной интерпретации языковых и жанровых вариантов необходимо учитывать «не только базовые экстралингвистические факторы коммуникации, но и более частные дискурсивные условия речевого общения» (Баженова, Ширинкина, 2022, с. 23). Дополнительное основание даёт Н. И. Клушина, которая подчёркивает методологическую значимость гипотезы об адресате, влияющей на выбор жанра, и рассматривает жанр как «горизонт ожиданий читателя», а интенционально-стилистический метод позволяет объяснить стилистический выбор и соотнести его с прототипической структурой жанра (Клушина, 2017, с. 62). В этом контексте стилистические неудачи рассматриваются как изменения, затрагивающие способ функционирования текста в профессиональной среде.

Стилистические неудачи фиксируются в тех случаях, когда перевод нарушает жанровую конвенцию документации или изменяет тональность таким образом, что текст перестаёт восприниматься как органичный элемент профессиональной коммуникации. Их специфика заключается в том, что они могут сохранять предметно-логическую сторону высказывания, но при этом изменять «режим» взаимодействия с адресатом: вводить неуместную экспрессию или, напротив, формировать излишнюю тяжеловесность и канцеляризм, не соответствующие жанру. В функциональном измерении такие решения влияют на восприятие текста и на его обучающую результативность; поэтому они описываются в терминах неудач, поскольку формальная нормативность может быть соблюдена при наличии функционального сбоя (неполучения «эквивалентного коммуникативного эффекта» в терминах коммуникативных подходов).

Выделение стилистических неудач как отдельного самостоятельного типа неудач связано также и с тем, что в документации норма задаётся профессиональным узусом, жанровыми ожиданиями и стилем конкретной платформы документации. Это напрямую выводит анализ на уровень адекватности: оценка требует учёта коммуникативной ситуации и адресатных ожиданий. В контексте стратегирования

стилистическая неудача часто выступает маркером непоследовательности в реализации выбранной линии перевода: переводчик может удерживать терминологическую точность, но теряет управляемость текста как инструмента обучения и сопровождения действий, что означает рассогласование внутри общего процесса планирования и контроля. Таким образом, стилистические неудачи выделяются по функциональному признаку: они снижают эффективность текста как профессионального коммуникативного инструмента при сохранении значительной части денотативного содержания.

1.5.5.4. Переводческие неудачи синтаксического характера

Переводческие неудачи синтаксического характера выделяются как особый тип неудач в силу того, что синтаксическая организация текста документации напрямую влияет на прозрачность инструкций, когнитивную доступность материала и управляемость чтения. В отличие от лингвоцентричности понимания синтаксической ошибки как нарушения грамматической (в отношении предложения) нормы, синтаксическая неудача в предлагаемой типологии фиксирует те случаи, когда синтаксический выбор формально допустим, но снижает функциональную результативность. Он может затруднять пошаговую интерпретацию, осложнять структуру условных и причинно-следственных связей, размывать условия применимости описываемого действия, нарушать привычные для жанра модели членения информации. Подобные решения могут оказываться критичными в жанре текста, ориентированном на выполнение действий адресатом а потому должны быть соотнесены с критериями качества, связывающими оценку перевода с успешностью

предметной деятельности пользователя, а не только с формальной организацией текстов.

С методологической точки зрения такие неудачи логично выводятся из функционального понимания качества и из стратегической природы переводческой деятельности. Если стратегирование мыслится как планирование и управление процессом с удержанием генеральной линии и возможностью корректировки решений, то синтаксическая организация текста перевода выступает одним из ключевых средств реализации этой линии на уровне макроструктуры высказывания и последовательности пошаговых инструкций. Отсюда следует, что неудача синтаксического характера может быть интерпретирована как сбой стратегического дизайна текста. Переводчик удерживает содержание и терминологию, однако теряет доступность структурирования знания, что снижает адекватность текста по отношению к коммуникативной ситуации профессионального характера и жанровым ожиданиям профессиональной среды. Выделение данного типа переводческих неудач, таким образом, мотивировано не формальным уровневым принципом, а функциональным. Синтаксис рассматривается при этом как механизм организации трансфера знаний и управления действиями адресата.

Выводы по главе 1

В первой главе представляются теоретические основания исследования перевода англоязычной документации по веб-разработке на русский язык. Проведённый анализ показал, что данная проблематика должна рассматриваться в рамках более широкой исследовательской перспективы, включающей трансфер

знаний, профессиональную коммуникацию, специфику технического текста и функционально ориентированный перевод. В условиях стремительного обновления цифровой среды именно документация становится одним из основных средств передачи и освоения специального знания, а её перевод обеспечивает включение этого знания в русскоязычную профессиональную практику.

Рассмотрение трансфера знаний в ИТ-сфере позволяет установить, что он представляет собой комплексный процесс, включающий сохранение, структурирование, адаптацию и практическое освоение информации. Веб-разработка занимает в этом процессе особое место, поскольку характеризуется высокой скоростью изменений, постоянным появлением новых технологий и необходимостью быстрого включения специалистов в новые профессиональные контексты. В такой ситуации документация выступает одним из центральных механизмов трансфера знаний, наряду с код-ревью, менторством, сообществами разработчиков, репозиториями и иными формами обмена опытом.

В ходе анализа показывается, что документация по веб-разработке является особой разновидностью технической документации, сочетающей в себе справочную, инструктивную и обучающую функции. Её задача заключается в объяснении принципов работы технологии, описании её функциональных возможностей, предоставлении точных инструкций и формировании у адресата способности использовать новый инструмент в практической деятельности. Для неё характерны структурированность, логическая организация материала, активное использование примеров кода, гиперссылок, визуальных элементов, типичных сценариев применения и пояснительных блоков. Всё это позволяет рассматривать документацию по веб-разработке как текст, ориентированный на предсказуемое действие пользователя и на усвоение технологии как системы взаимосвязанных знаний.

Анализ научной литературы по техническому тексту позволяет сделать вывод о том, что документация по веб-разработке сохраняет основные свойства технических

текстов, включая терминологическую организованность, логичность, точность, сжатость, однозначность, нормированность и высокую информационную насыщенность. Вместе с тем документация совмещает признаки технического и учебного текста. Это проявляется в особой роли пошаговых инструкций, примеров кода, пояснений, гипертекстовой навигации и доступного изложения, рассчитанного на адресатов разного уровня подготовки. На этом основании документация по веб-разработке определяется как особая разновидность специального текста, в которой передача знания тесно связана с организацией практической деятельности адресата.

Перевод документации по веб-разработке требует от переводчика глубокого понимания предметной области. Приоритетное значение приобретают точная передача терминологии, работа с многозначными и безэквивалентными единицами, структурно-смысловая адаптация текста, а также учёт жанровых и профессиональных ожиданий адресата. В подобных текстах качество перевода определяется не столько правильностью отдельных соответствий, сколько способностью текста сохранять прозрачность, инструктивность, логическую последовательность и пригодность для практического применения.

В целях понимания и определения сути переводческих неудач разграничиваются понятия переводческой ошибки и переводческой неудачи. Представленный обзор литературных источников показывает, что понятие переводческой ошибки традиционно используется для описания локальных отклонений перевода от оригинала, выявляемых в разных языковых аспектах. Однако применительно к документации по веб-разработке данное понятие оказывается недостаточным для оценки сути выявляемых неудач, поскольку часть из них не проявляется как очевидная языковая ошибка, но при этом снижается коммуникативная эффективность текста, ослабляется его обучающая направленность, нарушаются жанровые ожидания адресата, что может воспрепятствовать трансферу специальных знаний. В связи с этим понятие переводческой неудачи представляется более удачным для осуществления анализа в рамках данного исследования.

Методологическое обоснование такой позиции связано с применением идей интегративного переводоведения и коммуникативно-функциональной модели процесса перевода. Первый подход позволяет учитывать языковые, когнитивные, прагматические, дискурсивные и иные факторы, влияющие на переводческий результат. Второй акцентирует зависимость качества перевода от конкретной коммуникативной ситуации, цели перевода и условий его функционирования. С учётом этого переводческая неудача понимается как проявление рассогласования между функцией исходного текста и тем способом, которым она реализована в переводе.

Значимым для дальнейшего анализа представляется обращение к понятию переводческого стратегирования. Перевод может рассматриваться как процесс удержания общей линии решений, прогнозирования условий использования текста и коррекции переводческого курса в зависимости от цели, адресата и профессиональной среды. Это позволяет интерпретировать переводческие неудачи как последствие стратегических сбоев, то есть как результат непоследовательности или функциональной несогласованности переводческих решений.

2. Документация к ключевым технологиям веб-разработки: React JS, Vue.js и MDN Web Docs

Настоящая глава посвящена анализу документации трех фундаментальных ресурсов в области веб-разработки – библиотеки React JS, фреймворка Vue.js и справочника MDN Web Docs – с точки зрения их исторического развития, роли в индустрии и структурно-стилистических особенностей. Рассматриваются этапы эволюции каждой технологии, её вклад в формирование современных подходов к созданию пользовательских интерфейсов и веб-приложений, а также организация официальных материалов, включая англоязычные оригиналы и русскоязычные переводы. Особое внимание уделяется проблемам локализации, таким как актуальность, стилистическая однородность и терминологическая точность, что позволяет выявить общие вызовы перевода технической документации и их влияние на доступность знаний для русскоязычных специалистов.

2.1. React JS: краткая история развития, значимость в веб-разработке и структура документации

React JS представляет собой одну из фундаментальных библиотек в сфере создания веб-приложений, ориентированную на конструирование динамичных интерфейсов для взаимодействия с пользователями (Антонов и др., 2024; Нефёдов, Шехирев, 2025; Ратушняк, 2025). Возникновение данной библиотеки датируется 2011 годом, когда инженер Джордан Уолк разработал ее прототип с целью упрощения процессов управления содержимым в сложных онлайн-системах. Основная

концепция новой технологии заключалась изначально в разделении интерфейса на модульные, повторно используемые элементы, именуемые компонентами, что существенно отличалось от традиционных методик, предполагающих прямое вмешательство в структуру страницы. В период, когда веб-ресурсы эволюционировали в направлении большей сложности, с увеличением объемов данных и форм взаимодействия, существующие инструменты не всегда обеспечивали необходимую эффективность и масштабируемость. Эта инновация была обусловлена необходимостью оптимизации производительности в условиях растущих требований к скорости и отзывчивости приложений (Hamori, 2024).

Изначально React применялся в рамках крупных цифровых платформ для обеспечения оперативного обновления информации без заметных задержек для конечного пользователя. Центральным элементом нововведения стал механизм виртуальной модели документа, представляющий собой упрощенную копию реальной структуры страницы, которая позволяла выявлять и корректировать лишь необходимые изменения, избегая полной реконструкции. Такой подход значительно повысил быстродействие пользовательских интерфейсов, особенно в сценариях с частыми модификациями данных, таких как динамические списки или интерактивные элементы ввода, то есть структуры, которые автоматически обновляются при изменении данных (например, список сообщений в чате), и элементы управления, позволяющие пользователю вводить или изменять информацию в реальном времени (например, поля ввода, выпадающие меню, переключатели) (Komperla et al., 2022). В 2013 году библиотека React JS была официально представлена на конференции JavaScript Conference North America (сокр. JSConf US) (Web Design Museum), а ее исходный код открыт для общественности под лицензией Массачусетского технологического института (MIT), что стимулировало коллективные усилия по ее доработке и адаптации к различным нуждам разработчиков. Этот шаг ознаменовал переход от внутреннего инструмента к

глобально признанному стандарту в области веб-разработки (History and Evolution of React).

Дальнейшее развитие React характеризовалось серией значимых этапов, отражающих адаптацию к эволюционирующим потребностям отрасли. В 2014 году для таких этапов разработки пользовательского интерфейса, как анализ и отладка, были интегрированы специализированные инструменты, позволяющие мониторить элементы интерфейса в реальном времени непосредственно в веб-браузере. Годом позже, в 2015-м, появилась вариация React JS под названием React Native, расширяющая применение принципов компонентного подхода к мобильным приложениям для платформ iOS и Android. Хотя React Native ориентирован преимущественно на мобильную среду, его введение продемонстрировало универсальность исходной библиотеки в контексте кросс-платформенных решений, то есть подходов, позволяющих использовать один и тот же код для работы на разных устройствах и операционных системах (Taylor, 2025). В 2016 году 15 версия React JS внесла улучшения в механизмы обработки исключений, то есть встроенные способы перехвата и корректного реагирования на ошибки во время выполнения кода, и ввела поддержку генерации страниц на стороне сервера, что оказалось критически важным для оптимизации видимости в поисковых системах и ускорения начальной загрузки контента (Hamori, 2024).

Особо отметим трансформацию React JS в 2017 году благодаря выпуску React 16, известной как React Fiber, которая представляет собой полностью переработанную систему рендеринга, вводящую приоритетное распределение задач для повышения отзывчивости приложений к действиям пользователя (Patel, 2025). Напомним, что рендеринг – это процесс превращения описания интерфейса в его видимую форму на экране, когда абстрактная схема элементов переводится в реальные кнопки, списки или изображения, с которыми взаимодействует человек (Paramadani et al., 2023). Данная архитектура заложила фундамент для параллельных процессов обработки интерфейсов, минимизируя блокировки и обеспечивая более

плавный пользовательский опыт. В 2018 году обновление React JS 16.8 представило концепцию «хуков» (функциональных инструментов, таких как *useState* для хранения информации и *useEffect* для реагирования на изменения), которые упростили структуру кода, сделав его более интуитивным и менее зависимым от сложных конструкций. Это нововведение снизило порог входа для начинающих специалистов, способствуя дальнейшему распространению React JS. К 2020 году 17-я версия React JS сфокусировалась на обеспечении совместимости с предшествующими релизами, то есть более ранними официальными выпусками программы, чтобы разработчики могли безболезненно обновлять проекты и использовать новые возможности без нарушения работы существующего кода. А созданная в 2022 году React 18 ввела полноценные механизмы параллельной отрисовки, то есть процесса отображения элементов интерфейса на экране, позволяющего системе одновременно обрабатывать несколько задач. В эту систему были встроены переходы и задержки – специальные приёмы, которые помогают сгладить изменения интерфейса и управлять временем отклика при асинхронных операциях. Благодаря им пользователь видит плавные обновления (например, при загрузке новых данных или переключении между страницами), а приложение остаётся отзывчивым даже в сложных сценариях обработки информации (Ali, 2025).

В период с 2023 по 2025 гг. эволюция React продолжилась в направлении интеграции системы с передовыми технологиями. В 2023 году были усилены возможности работы с серверными компонентами, способствующие ускорению загрузки страниц за счет предварительной обработки на сервере (Bezverhiy, Kutsenko, 2024). К 2024 году акцент сместился на повышение совместимости React JS с общепринятыми веб-стандартами и улучшение доступности интерфейсов для пользователей с ограниченными возможностями. В 2025 году наблюдается тенденция к внедрению в библиотеку React JS элементов искусственного интеллекта, направленных на автоматизацию проверки и генерации элементов интерфейса (Orazgeldiyeva et al., 2025). Представленные этапы развития технологии React JS

иллюстрируют, как React адаптируется к изменяющимся парадигмам разработки, от монолитных структур к распределенным системам, обеспечивая устойчивость и актуальность в динамичной цифровой среде. Исследования подчеркивают, что такой прогресс способствует оптимизации кросс-платформенных приложений, минимизируя издержки и повышая производительность работы веб-разработчиков (там же).

Значение React JS в контексте веб-разработки невозможно переоценить, поскольку эта технология фундаментально преобразовала подходы к созданию интерфейсов. Согласно опросам, таким как State of JavaScript, библиотека стабильно лидирует среди инструментов для фронтенд-разработки, находя применение в миллионах проектов глобального масштаба (Why Choose React for Web Development in 2025). Компонентная модель позволяет конструировать масштабные приложения с использованием повторяемых фрагментов кода, что сокращает временные и ресурсные затраты на разработку пользовательских интерфейсов. Механизм виртуальной модели документа обеспечивает повышенную эффективность, особенно в системах с интенсивным потоком данных, включая платформы электронной коммерции или аналитические панели. Технология React стимулировала формирование экосистемы дополнительных библиотек, таких как Redux для централизованного хранения данных, React Router для навигации и Material-UI для стилизации, обогащая инструментарий разработчиков. Более того, её влияние распространилось на смежные фреймворки, вдохновив их тем самым на заимствование идей модульности и реактивности (Shogenov, 2023).

В рамках комплексной разработки React гармонично интегрируется с платформами вроде Next.js, позволяя создавать приложения с серверной обработкой, в которых приоритет отдается скорости загрузки и оптимизации поисковых механизмов. Это делает её оптимальным выбором для современных цифровых продуктов, в которых ключевыми факторами выступают производительность и удобство пользования (Sidorov, 2024). Кроме того, React способствует унификации

кода между веб-платформами и мобильными платформами, минимизируя дублирование усилий и повышая тем самым экономическую эффективность проектов (Kniazev, 2023). В глобальном аспекте библиотека React JS демократизировала доступ к профессиональной веб-разработке благодаря обширному сообществу и открытым ресурсам, что подтверждается данными репозиториев, то есть специализированных хранилищ исходного кода и его версий, вроде GitHub, с миллионами отметок и ветвлений. На рынке труда требования к владению React доминируют, подчеркивая её стратегическую роль в формировании компетенций специалистов. Научные работы отмечают преимущества React в создании масштабируемых архитектур, в которых компонентный подход обеспечивает гибкость и устойчивость к росту сложности систем (Garifulin, 2024; Антонов и др., 2024; Komperla et al., 2022).

Структура документации React организована на официальном ресурсе react.dev (ранее reactjs.org), выступающем в качестве основного справочного материала для аудитории различного уровня подготовки. Документация разделена на логические блоки, ориентированные на последовательное освоение. Раздел “Learn” («Изучение») предлагает пошаговое введение в основные принципы работы технологии, начиная от фундаментальных понятий, таких как формирование компонентов, и переходя к более сложным аспектам, включая управление данными и навигацию. Здесь предусмотрены интерактивные примеры, редактируемые в браузере, что способствует отработке практических навыков. Справочник API детализирует встроенные функции и элементы, группируя их по категориям для удобства поиска. Отдельное внимание уделено «хукам» как центральному механизму современной версии React JS. Раздел “Communities” («Сообщества») предоставляет ссылки на форумы, публикации и внешние дополнения, стимулируя коллективное взаимодействие (React JS. Официальная документация).

В дополнение к основному руководству включены блог для анонсов обновлений. Общая архитектура иерархична: от общих обзоров к более конкретным

нюансам, к инструментам поиска, навигации и выбора версий. Дизайн лаконичен, с поддержкой тем в виде визуальных элементов, таких как диаграммы. Объем материалов исчисляется сотнями страниц, с акцентом на практическое применение (там же).

Официальный перевод документации к React JS на русский язык доступен по адресу ru.react.dev и в целом воспроизводит структуру англоязычного оригинала. Он осуществляется силами волонтеров через платформу GitHub, на которой возможны коллективные корректировки. Содержание охватывает ключевые разделы по обучению и справочнику, сохраняя примеры в исходном виде для универсальности. Стандартизация терминов, таких как «компонент» или «состояние», способствует сохранению единообразия документации. Однако на данный момент перевод уже устарел и, на наш взгляд, требует существенной доработки и обновления. Задержки в синхронизации с новыми разделами оригинала приводят к неполноте текста, а вариации в формулировках могут вводить в заблуждение. Это особенно актуально для описаний инновационных функций, в которых теряются важные нюансы, подчеркивающие необходимость регулярного аудита и актуализации локализованных материалов (React JS. Официальная документация).

Таким образом, React JS развился из внутреннего прототипа в один из основных стандартов создания современных веб-интерфейсов, заложив принципы разделения интерфейса на повторно используемые блоки (компоненты), эффективного обновления страницы и автоматической реакции на изменения данных. Его значение определяется широким применением в индустрии, поддержкой множества дополнительных инструментов и высоким спросом на специалистов, владеющих этой технологией. Официальная документация на сайте react.dev организована последовательно: от базовых понятий до детального описания функций, с примерами и справочными материалами. Русскоязычная версия (ru.react.dev) сохраняет общую структуру и единую терминологию, однако содержит устаревшие разделы, неполные переводы и задержки в обновлениях. Это создаёт

трудности для русскоязычных разработчиков и подчёркивает необходимость регулярного обновления перевода силами сообщества, чтобы обеспечить равный доступ к актуальным знаниям и поддержать профессиональное развитие специалистов.

2.2. Vue.js: краткая история развития, значимость в веб-разработке и структура документации

Vue.js представляет собой прогрессивный JavaScript-фреймворк, ориентированный на создание пользовательских интерфейсов и разработку одностраничных веб-приложений (Байдыбеков, 2020). Он был создан в 2014 году разработчиком Эваном Ю, который ранее работал в компании Google над проектом AngularJS. В отличие от существовавших на тот момент инструментов, технология Vue была задумана как более лёгкое, гибкое и доступное решение, объединяющее сильные стороны крупных фреймворков и простоту «чистого» языка программирования JavaScript. Исходная идея заключалась в создании инструмента, который позволил бы постепенно внедрять реактивность, то есть автоматическое обновление интерфейса при изменении данных, в уже существующие страницы, не требуя их полной перестройки. Эта концепция постепенной интеграции определила архитектурную особенность Vue и оказала значительное влияние на его популяризацию в профессиональном сообществе (Emmanni, 2023; Яровая 2022).

Первые версии Vue отличались небольшим размером, минимальным порогом входа и ясной структурой, что способствовало их быстрому распространению среди разработчиков малых и средних проектов. Поддержка шаблонов на основе HTML,

прозрачная система реактивности и декларативный подход к описанию логики интерфейса оказались востребованы в условиях роста требований к интерактивности веб-приложений. Реактивная модель данных Vue основана на отслеживании зависимостей, то есть автоматическом выявлении связей между данными и элементами интерфейса, и их синхронном обновлении при изменении состояния, что существенно упрощало поддержку динамических компонентов и делало процесс разработки предсказуемым и контролируемым (Vue.js. Официальная документация).

К 2016 году Vue достиг широкой известности благодаря выходу версии 2.0. В ней была полностью переработана внутренняя система рендеринга, основанная на виртуальной модели документа. Это позволило повысить производительность при работе со сложными интерфейсами и обеспечило корректную обработку изменений в больших наборах данных. Реактивность была реализована на основе геттеров и сеттеров JavaScript, то есть специальных методов для чтения и изменения свойств объекта, которые позволяют автоматически отслеживать изменения данных и запускать обновление интерфейса, что обеспечивало высокую скорость обновления отображаемых данных и делало Vue эффективным инструментом для приложений с интенсивными пользовательскими операциями. В этот период Vue стал активно использоваться в крупномасштабных проектах, включая корпоративные панели управления, сервисы электронной коммерции и приложения с большим количеством взаимодействий (Матвеев, 2020).

Значение Vue в современной веб-разработке определяется его универсальностью и адаптивностью. Фреймворк используется как в небольших интерфейсных модулях, встроенных в существующие серверные приложения, так и в полномасштабных одностраничных системах, созданных на основе инструментов Vue CLI или Vite. Благодаря интуитивно понятному синтаксису, постепенной обучаемости и широкому сообществу Vue остаётся одним из наиболее популярных решений, что подтверждается регулярными результатами отраслевых опросов и

статистикой репозиториев открытого исходного кода (Яровая, 2022; Матвеев, 2020; Антонов и др., 2024).

Исторически документация Vue развивалась параллельно с развитием самого фреймворка (Vue.js. Официальная документация). В версии 1.0 основное внимание уделялось базовым концепциям, изложенным в виде кратких разделов, ориентированных на демонстрацию принципа *progressive enhancement* – постепенного усложнения проекта. Авторы стремились раскрыть возможности фреймворка так, чтобы он был понятен начинающим пользователям, не обладающим глубокими знаниями в области JavaScript. С выходом Vue 2 документация стала гораздо более разветвлённой и структурированной, появились подробные руководства, расширенный справочник API (Application Programming Interface – набор доступных функций и параметров), пояснения к примерам, а также более чёткая логическая организация материалов. Эта эволюция сопровождалась значительным увеличением объёма текста и повышением требований к последовательности изложения. Начиная с Vue 3, выпущенного в 2020 году, структура документации снова была пересмотрена: появились дополнительные объяснительные разделы, развёрнутые примеры и отдельные страницы, предназначенные специально для обучающих целей. Таким образом, документация Vue стала своеобразным учебно-методическим комплексом, описывающим способы мышления, характерные для современного фронтенд-разработчика.

Структура документации Vue.js представлена на официальном сайте vuejs.org и служит систематизированным источником знаний для разработчиков с разным уровнем подготовки. Основной раздел “Guide” («Руководство») выстроен как последовательное введение: от базовых принципов реактивности и построения компонентов до более сложных тем, связанных с маршрутизацией и управлением состоянием. Важной особенностью является наличие интерактивных примеров, позволяющих проверять работу кода непосредственно в браузере. Раздел “API Reference” («Справочник API») содержит подробное описание встроенных функций,

директив и методов, сгруппированных для удобного поиска. Дополнительно выделен блок *Ecosystem*, где представлены официальные библиотеки, такие как Vue Router и Vuex, а также ссылки на плагины и ресурсы сообщества, что подчёркивает ориентацию Vue на расширяемость и коллективное развитие (Vue.js. Официальная документация).

Официальная документация существует на английском языке, однако создатели Vue поддерживают инициативу по переводу на другие языки, включая русский. При этом важно отметить, что русская версия документации не всегда обновляется синхронно с оригиналом: локализация осуществляется сообществом, и её актуальность зависит от активности участников. В отличие от некоторых технологий, где официальный перевод поддерживается корпорациями и отличается строгой нормированностью, перевод Vue создаётся энтузиастами, что делает его более гибким, но одновременно подверженным вариативности уровня качества. Это обстоятельство значительно влияет на характер текста: перевод иногда оказывается стилистически неоднородным, не полностью соответствующим оригиналу по тону и степени формальности, а терминология может демонстрировать определённые расхождения.

Стиль документации Vue построен на принципе прозрачности и диалогичности. Текст часто апеллирует к читателю, предлагает вопросы и сравнения, стимулирует самостоятельное изучение материала. Преобладают простые синтаксические конструкции, активный залог и глагольные формы, отражающие действие. В русском переводе часть этих особенностей утрачивается: сложность грамматического строя русского языка нередко приводит к удлинению фраз или использованию пассивных конструкций, что может уменьшать ощущение непосредственного взаимодействия с текстом. Кроме того, оригинальные метафоры или лёгкие языковые игры могут оказаться непростыми для перевода или требовать адаптации, что создаёт риск стилистических отклонений.

К достоинствам русскоязычного перевода можно отнести стремление сохранить точность терминологии и структурную логику изложения. Перевод создаётся при участии достаточно широкого круга энтузиастов, многие из которых являются разработчиками, что обеспечивает высокий уровень технической корректности. Кроме того, перевод зачастую адаптируется в соответствии с актуальными нормами русского технического стиля, что делает его более привычным для тех, кто читает российские пособия и учебные материалы. Однако недостатком может быть отсутствие единообразия стиля: разные разделы нередко переводятся разными людьми, и редакторская вычитка не всегда обеспечивает полное стилистическое согласование. Это приводит к наличию фрагментов с различной степенью формальности, разными подходами к передаче терминов и отличиями в структуре предложений. Ещё одним недостатком является то, что русская версия документации иногда отстаёт от английской по обновлениям, что особенно критично в условиях быстрого развития экосистемы Vue. Обновления английского оригинала иногда вводят новые концепции, изменяют устаревшие элементы или корректируют рекомендуемые практики, тогда как русская версия может сохранять устаревшие формулировки, что снижает её обучающую ценность.

Стилистические особенности документации Vue также требуют внимательного анализа с точки зрения переводоведения. Английский текст характеризуется высокой степенью эксплицитности (т. е. прямого объяснения), что облегчает понимание, но создаёт трудности при переводе в языковую систему, в которой образовательные тексты часто строятся иначе – с опорой на более компактные формулировки и логические обобщения. Кроме того, в оригинале используются элементы повествовательности, включая сценарии «предположительного опыта» («представьте, что вы хотите…», «если вы столкнулись с ситуацией…» (Vue.js. Официальная документация)), которые не всегда естественны для русской технической традиции. Переводчики часто стремятся к большей стандартизации и устранению эмоционально-оценочных элементов, что

делает русскую версию более строгой, но менее привлекательной для новичков. Некоторые особенности оригинального текста, например выражения, направленные на снятие психологического барьера у начинающих пользователей («не волнуйтесь, всё просто», «если что-то кажется сложным, мы сейчас это разберём» (там же)), могут выглядеть чрезмерно разговорными в русском контексте. В результате перевод зачастую отказывается от них или заменяет на более обобщённые конструкции, что уменьшает педагогическую эффективность текста.

Таким образом, технология Vue.js эволюционировала от простого инструмента для постепенного добавления интерактивности к полноценному фреймворку для создания сложных одностраничных приложений, сохраняя при этом простоту и гибкость, что обеспечило ей широкую популярность в проектах различного масштаба. Её значимость определяется интуитивным синтаксисом, эффективной реактивностью и активным сообществом, поддерживающим экосистему дополнительных библиотек. Официальная документация на vuejs.org организована как последовательное руководство с интерактивными примерами, справочником API и разделами по экосистеме, сочетая техническую точность с доступным изложением. Русскоязычная версия, создаваемая энтузиастами, сохраняет структуру и терминологию, но страдает от задержек в обновлениях, стилистической неоднородности и частичной утраты диалогического тона оригинала.

2.3. MDN Web Docs: краткая история развития, значимость в веб-разработке и структура документации

MDN Web Docs (ранее известный как Mozilla Developer Network, MDN) представляет собой крупнейший в мире открытый справочник по веб-технологиям, созданный при участии Mozilla и поддерживаемый широкой международной общиной разработчиков, технических писателей и добровольцев; ресурс объединяет справочные материалы, учебные руководства и примеры кода по базовым и продвинутым технологиям веба, таким как HTML (язык структуры веб-страниц), CSS (язык описания внешнего вида) и JavaScript (язык программного поведения), а также по множеству сопутствующих интерфейсов и API, то есть набору интерфейсов программирования, через которые веб-страница взаимодействует с браузером и устройством (MDN Web Docs). MDN выполняет роль канала трансфера знаний, то есть обеспечивает передачу практического и концептуального знания от создателей стандартов и опытных разработчиков к широкому кругу пользователей – от студентов и специалистов начального уровня до профессионалов, которым необходимы точные спецификации.

Проект возник как инициатива Mozilla в середине 2000-х годов и вначале представлял собой платформу для документирования открытых веб-стандартов и технологий, связанных с браузером Firefox; в 2010-х годах ресурс был переосмыслен и переформатирован в MDN Web Docs, а в 2017 году Mozilla объявила о стратегическом сотрудничестве с другими крупными участниками отрасли (включая Google и Microsoft) с целью консолидировать документацию по веб-API и повысить единообразие информации для разработчиков во всём мире (MDN Web Docs). Такое объединение сделало MDN центральной точкой доступа к официальной документации многих веб-технологий, поскольку крупные вендоры начали направлять свои материалы и усилия по документации в сторону единого, открытого репозитория (Iffland, 2017).

Практическая значимость MDN в профессиональной среде обусловлена несколькими обстоятельствами. Во-первых, MDN сочетает два типа текстов: справочный (короткие, точные описания функций, методов и их параметров) и

учебный (пошаговые руководства, объяснения и демонстрационные примеры). Такое сочетание делает ресурс полезным и для быстрого справочного поиска, и для системного обучения. Во-вторых, MDN систематически собирает данные о поддержке конкретных возможностей в разных браузерах (таблицы совместимости), что критично для практического использования функций в реальных проектах: разработчик может быстро увидеть, в каких версиях браузеров тот или иной метод доступен (MDN Web Docs). В-третьих, MDN – открытый проект с публичными репозиториями и инструкциями по внесению правок; это позволяет сообществу быстро исправлять ошибки, дополнять примеры и актуализировать материалы в ответ на эволюцию стандартов. В совокупности эти факторы делают MDN одним из наиболее влиятельных посредников в трансфере практических знаний о веб-технологиях: информация, сформулированная и отредактированная на MDN, регулярно используется в учебных курсах, коммерческих проектах и как ориентир при принятии инженерных решений (Schafer, 2017).

Структура MDN Web Docs построена таким образом, чтобы балансировать между обучением и справкой. На уровне сайта выделяют крупные тематические разделы: “Web” (общее представление о веб-технологиях), “Learn” (обучающие курсы и путеводители для начинающих), “Reference” (справочники по HTML, CSS, JavaScript и Web APIs) и вспомогательные разделы об экосистеме и рекомендациях по написанию контента (MDN Web Docs). В обучающих разделах текст обычно даёт постепенные вводные объяснения, сопровождаемые практическими заданиями и интерактивными примерами. Это рассчитано на читателя-непрофессионала – человеку, не знакомому с программированием, такие вводные дают образный уровень понимания, например «HTML – это набор тегов, которые создают структуру страницы» (там же). Справочные страницы ориентированы на быструю ориентировку: они следуют шаблону «определение → синтаксис → параметры → примеры → совместимость браузеров», что облегчает нахождение требуемой технической информации.

Текстовый стиль MDN Web Docs во многом схож с текстами документации к иным технологиям веб-разработки: авторы MDN поддерживают «говорящий», доступный тон, нередко обращаются к читателю напрямую (например: “you can…”, “let’s try…” (MDN Web Docs)) и используют наглядные метафоры и простые примеры для объяснения абстрактных понятий. Одновременно справочные записи стремятся к нейтральной, безэмоциональной формулировке фактов, что требует точной терминологии. Наличие двух таких пластов – дидактического и справочного – позволяет ставить документацию MDN Web Docs в один ряд с текстами документации к React JS и Vue.js: тексты должны быть и обучающими, и технически точными и корректными.

Вопрос локализации у MDN решается в рамках открытой модели: существует инфраструктура и репозитории для перевода, есть официально поддерживаемые языковые версии, и перевод осуществляется сообществом добровольцев при участии кураторов и поддержке Mozilla. На момент написания данной работы проект поддерживает несколько активных языковых версий, в том числе русский; локализация включает как перевод учебного материала, так и справочных страниц. Перевод организован через публичные репозитории и обсуждения, где волонтёры могут предлагать правки и синхронизировать изменения с английским оригиналом.

Отметим, что у модели локализации есть и ограничение, аналогичное документации к React JS и Vue.js. Главная организационная трудность заключается в том, что переводы поддерживаются сообществом и потому иногда отстают от обновлений английской версии; это означает, что в русскоязычной части MDN может отсутствовать самая свежая информация о недавно появившихся возможностях или изменениях в рекомендациях. Кроме того, качество перевода может варьироваться: разные участники вносят правки с различным уровнем редакторской вычитки, и в отдельных случаях фиксируются ошибки или стилистические несоответствия, которые впоследствии обсуждаются и исправляются в репозиториях проекта. Примеры таких обсуждений и конкретных проблем доступны в публичных ишью на

GitHub, то есть в открытых карточках-обсуждениях, где фиксируются ошибки, предложения и задачи по развитию проекта (GitHub: Перевод MDN на русский язык #276).

Достоинства перевода MDN на русский язык, на наш взгляд, включают: 1) доступность и масштаб – большая часть основного корпуса материала по базовым технологиям действительно переведена и доступна; 2) сообщество перевода состоит из практикующих разработчиков, что повышает техническую корректность переведённых описаний; 3) инфраструктура (репозитории, гайды по переводам, правила стиля) обеспечивает стандартизированный подход к локализации. Среди очевидных недостатков находятся: 1) отставание локализации относительно оригинала, особенно в разделах, связанных с новыми спецификациями и современными API; 2) стилистическая неоднородность, вызванная коллективным характером перевода и разницей транслируемых языковых норм; 3) риск терминологической неоднозначности для читателей без технической подготовки (например, термины «реактивность», «компонент», «DOM» требуют пояснений и контекстуализации), что иногда реализуется не в полной мере (DMN Web Docs).

Таким образом, MDN Web Docs – крупнейший и ключевой открытый репозиторий по веб-технологиям, который выполняет роль центра трансфера практических знаний; его архитектура и редакционные принципы ориентированы на сочетание обучающего и справочного стилей, что делает ресурс одновременно пригодным для обучения и надёжным справочником. Перевод MDN Web Docs на русский язык существенен для расширения аудитории и демократизации доступа к знаниям, но локализация, выполняемая сообществом, несёт в себе как преимущества (широкая база волонтёров, техническая компетентность переводчиков, открытость правок), так и ограничения (отставание обновлений, вариативность качества, потребность в более тщательной редактуре для поддержания единого стиля и понятности непрофильному читателю). При использовании MDN русскоязычные реципиенты опираются на переведённые вводные и учебные материалы для базового

понимания, а при работе с новыми API и точными спецификациями – могут быть вынуждены дополнительно сверяться с английской версией или с оригинальными спецификациями стандартов.

Выводы по главе 2

Во второй главе была рассмотрена документация трёх ключевых ресурсов современной веб-разработки – React JS, Vue.js и MDN Web Docs – с точки зрения их исторического развития, значения для профессиональной среды и особенностей организации официальных материалов. Проведённый анализ показал, что каждый из этих ресурсов занимает особое место в системе веб-технологий и одновременно выступает значимым каналом трансфера специальных знаний, обеспечивая освоение новых инструментов, формирование профессиональных практик и распространение единых представлений о способах работы с веб-технологиями.

React JS охарактеризован как одна из фундаментальных библиотек для создания пользовательских интерфейсов, развитие которой связано с переходом веб-разработки к компонентному принципу организации интерфейса и к более эффективным механизмам обновления данных на странице. Анализ истории React JS показал, что библиотека прошла путь от внутреннего решения для сложных цифровых платформ до одного из отраслевых стандартов фронтенд-разработки. Её значимость определяется высокой производительностью, масштабируемостью, устойчивой экосистемой дополнительных инструментов и широким распространением в профессиональной среде. Рассмотрение документации React JS позволило установить, что она организована как последовательная система

обучающих и справочных материалов, ориентированных на пользователей разного уровня подготовки. При этом русскоязычная версия документации, сохраняя общую структуру оригинала, характеризуется отставанием в обновлениях, неполнотой отдельных разделов и вариативностью переводческих решений, что снижает её ценность как актуального источника профессионального знания.

Vue.js был рассмотрен как прогрессивный фреймворк, значение которого связано с его гибкостью, относительной простотой и возможностью постепенного внедрения в уже существующие веб-проекты. История Vue показывает, что его архитектурная модель формировалась как ответ на потребность в более доступном и менее громоздком инструменте для построения реактивных интерфейсов. В главе установлено, что важной чертой Vue стала его ориентация на последовательное освоение технологии, что нашло прямое отражение и в структуре официальной документации. Документация Vue организована как развернутое руководство, совмещающее объяснение базовых принципов, справочное описание API и материалы по экосистеме. При этом её стиль строится на прозрачности и диалогичности, что усиливает обучающую функцию текста. Анализ русскоязычной версии документации позволил выявить типичные для коллективной локализации особенности: относительную техническую точность при недостаточном стилистическом единообразии, неравномерности качества разных разделов и задержках в обновлении по сравнению с английским оригиналом.

MDN Web Docs в главе представлен как крупнейший открытый справочник по веб-технологиям, выполняющий функцию универсального репозитория знаний по HTML, CSS, JavaScript, Web API и другим технологиям веб-платформы. В отличие от документации React и Vue, ориентированной на конкретные инструменты, MDN охватывает базовую инфраструктуру веб-разработки в целом. Это определяет его особый статус в профессиональной среде. Анализ показал, что MDN сочетает признаки справочника и учебного ресурса: с одной стороны, он предоставляет точные описания синтаксиса, параметров и совместимости браузеров, с другой

предлагает систематизированные обучающие материалы и примеры для пользователей разного уровня подготовки. Существенной особенностью ресурса является открытая модель разработки и локализации, благодаря которой документация постоянно дополняется и уточняется международным сообществом. Русскоязычная версия MDN расширяет доступ к профессиональному знанию, однако также демонстрирует характерные ограничения коллективного перевода: отставание от англоязычного оригинала, стилистическую неоднородность и необходимость дополнительной редакторской вычитки в ряде разделов.

Рассмотрение React JS, Vue.js и MDN Web Docs позволяет установить, что все три ресурса объединяет общая функция передачи и структурирования специальных знаний в цифровой среде, однако реализуется она по-разному. Документация React JS в большей степени ориентирована на освоение конкретной библиотеки и связанных с ней паттернов разработки. Документация Vue совмещает объяснение инструмента с выраженной учебной логикой и постепенным вовлечением пользователя в принципы реактивной разработки. MDN Web Docs, в свою очередь, выполняет роль широкого справочно-образовательного пространства, обеспечивающего системное представление о веб-платформе как таковой. Таким образом, различия между этими ресурсами касаются характера коммуникативной организации материала, степени диалогичности, структуры объяснения и соотношения учебных и справочных компонентов.

Проведённый анализ позволяет утверждать, что официальная документация в области веб-разработки в современных условиях не может рассматриваться исключительно как вспомогательный технический материал. Она выступает в качестве полноценного инструмента профессиональной социализации, обеспечивая доступ к терминологии, практикам использования технологии, типовым сценариям работы и принятым в сообществе способам представления знания. Именно поэтому её качество, структура, стилистическая согласованность и актуальность оказываются критически важными как для англоязычного оригинала, так и для перевода. В случае

русскоязычных версий документации особую значимость приобретают синхронность обновления, единообразие терминологии и редакторская выверенность, поскольку любые расхождения между оригиналом и переводом непосредственно влияют на доступность и точность передаваемого знания.

Подводя итог, следует отметить, что React JS, Vue.js и MDN Web Docs представляют собой значимые формы организации специального знания в сфере веб-разработки. Их документация отражает разные модели подачи профессиональной информации, разные способы сочетания обучающей и справочной функций и разные подходы к коллективной локализации. Рассмотрение этих ресурсов позволило выявить те свойства документации, которые оказываются существенными для последующего переводоведческого анализа: зависимость качества перевода от актуальности оригинала, важность стилевого и терминологического единообразия, чувствительность текста к способу организации материала и особую роль локализованной документации в обеспечении доступа русскоязычных специалистов к современным веб-технологиям. Именно эти наблюдения создают необходимую основу для перехода к практическому анализу переводческих неудач в третьей главе исследования.

3. Анализ переводческих неудач в переводе англоязычной документации по веб-разработке на русский язык

3.1. Материал и методика исследования

Материалом настоящего исследования послужили англоязычные оригиналы и переводы на русский язык официальной документации к ключевым технологиям веб-разработки: React JS, Vue.js и MDN Web Docs. Анализ проводился на основе сопоставления микроконтекстов, извлечённых из соответствующих версий документации, с целью выявления и систематизации переводческих неудач, возникающих в процессе трансфера знаний в сфере веб-разработки.

Общий объём корпуса языкового материала составил 1,8 МБ, содержащих параллельные (английские и русские) микроконтексты, включающие в себя в общей сложности 527 переводческих неудачи, выявленные в результате последовательного анализа англоязычных и русскоязычных версий документации.

Наибольшее количество переводческих неудач зафиксировано в документации React JS, на долю которой приходится 54,84% корпуса (289 микроконтекстов). Существенно меньшая часть материала относится к документации MDN Web Docs (31,31%, 165 микроконтекстов), и наименьшая – к документации Vue.js (13,85%, 73 микроконтекста) (см. Рисунок 6).

Значительное преобладание переводческих неудач в документации React JS не может быть объяснено исключительно объёмом анализируемого материала или спецификой представленности жанра. Существенную роль в формировании данного распределения играет характер используемой переводческой стратегии и степень

последовательности её реализации, а также организация перевода русскоязычной версии документации.

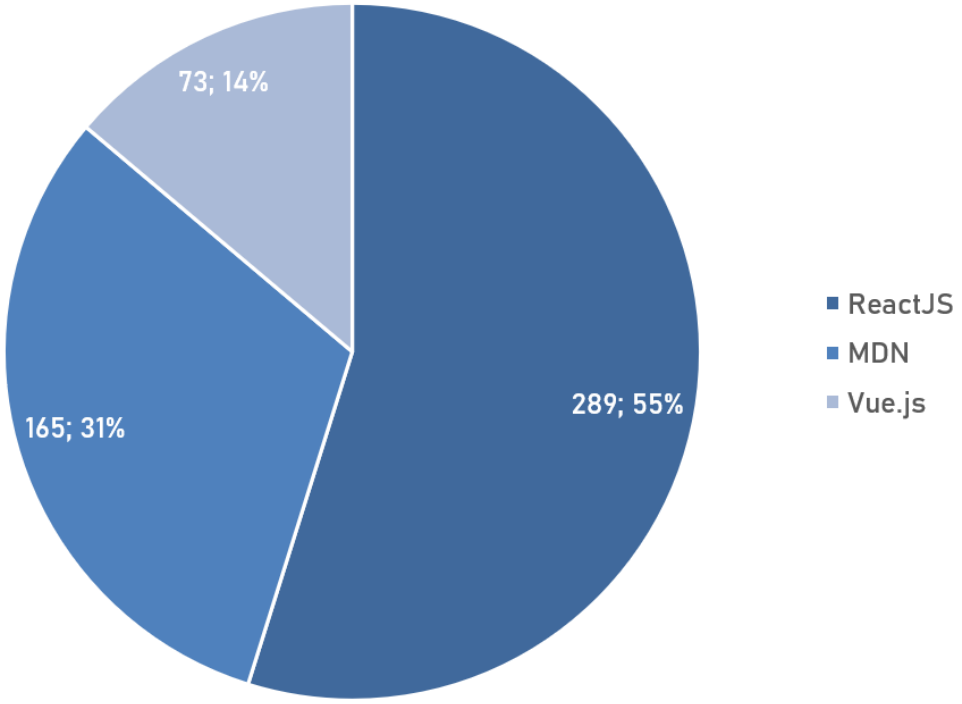


Рисунок 6. Распределение переводческих неудач по документациям

Анализ корпуса показывает, что перевод документации React JS отличается менее последовательным подходом к воспроизведению коммуникативных и прагматических параметров оригинального текста. В частности, в переводе систематически наблюдаются добавления, сглаживание модальной семантики, ослабление директивно-инструктивных элементов и неоднородность в передаче метаязыка документации. Подобные особенности приводят к накоплению переводческих неудач даже в тех случаях, когда предметное содержание микроконтекстов формально сохраняется, а расхождения проявляются преимущественно на уровне коммуникативно-прагматической организации текста.

Дополнительным фактором, усиливающим данный эффект, является асинхронность обновления оригинальной и переводной версий документации React JS. Обновлённая версия англоязычной документации, размещённая на сайте

react.dev, на момент написания настоящей работы не имеет полного перевода на русский язык. В результате русскоязычные разработчики вынуждены опираться на более раннюю версию документации, перевод которой в ряде случаев не отражает ни актуальную структуру оригинального материала, ни его современную коммуникативную организацию. Это обстоятельство создаёт благоприятные условия для возникновения переводческих неудач и усиливает их системный характер.

В противоположность этому, перевод документации MDN Web Docs и Vue.js демонстрирует более высокий уровень переводческой проработки. Для данных ресурсов характерна большая последовательность в передаче модальных и инструктивных элементов, более стабильное воспроизведение терминологии и более точное следование логике оригинального текста. В результате количество выявленных переводческих неудач в данных текстах документации оказывается существенно ниже, а сами неудачи реже повторяются и в меньшей степени носят системный характер.

Следует подчеркнуть, что более низкая частотность переводческих неудач в документациях MDN Web Docs и Vue.js не означает их отсутствие. Однако выявленные в них неудачи, как правило, носят локальный характер и характеризуются меньшей повторяемостью, в отличие от документации React JS, в которой многие неудачи проявляются системно и затрагивают однотипные микроконтексты.

В качестве источника англоязычного оригинала для библиотеки React JS использовалась официальная документация, размещённая на сайте legacy.reactjs.org. В качестве материала перевода на русский язык анализировалась документация, доступная на сайте ru.legacy.reactjs.org, которая воспроизводит структуру и содержание более ранней версии англоязычного оригинала.

В качестве англоязычного оригинала документации Vue.js использовались материалы с официального сайта vuejs.org, а перевод представлен русскоязычной версией того же ресурса.

Для анализа MDN Web Docs использовались страницы англоязычной версии сайта developer.mozilla.org и соответствующие страницы русскоязычной версии ресурса. Во всех случаях материалом исследования являлись официальные версии документации, поддерживаемые или признаваемые сообществом разработчиков.

В рамках исследования в качестве единицы анализа использовался микроконтекст, представляющий собой минимальный фрагмент документационного текста, обладающий относительной смысловой и прагматической завершённостью и достаточный для выявления переводческой неудачи. В качестве микроконтекстов рассматривались отдельные предложения и абзацы, заголовки, пояснительные элементы инструкций, а также сопровождающие комментарии к примерам кода.

Материал исследования включает вербальные элементы текста и встроенные элементы гипертекста: фрагменты кода с пояснениями, гиперссылки, элементы визуального выделения (полужирное начертание, списки, структурные акценты), а также навигационные элементы, интегрированные в текст документации. Анализ данных элементов осуществлялся в тех случаях, когда особенности их перевода или графического оформления оказывали влияние на интерпретацию микроконтекста и на функциональную состоятельность перевода.

Отбор микроконтекстов в корпус производился по принципу наличия функционального расхождения между оригиналом и переводом, не сводимого к формальному нарушению языковой нормы. В анализ включались только те случаи, в которых перевод сохранял предметное содержание оригинала, однако демонстрировал изменение или ослабление одной или нескольких коммуникативных функций документационного текста. Это позволило сосредоточиться на переводческих неудачах как на явлениях функционального порядка, а не на единичных языковых несоответствиях.

В ходе исследования применялся комплекс взаимодополняющих методов анализа. Сопоставительный анализ использовался для выявления различий между англоязычными и русскоязычными микроконтекстами. Осуществляемый лексико-

семантический анализ позволял фиксировать смещения в объёме и структуре значений лексем, включая интерпретирующие добавления и утрату смысловых компонентов. Прагматический анализ применялся для оценки степени реализации коммуникативной интенции оригинального текста и управляемости сценария действий реципиента в переводе. Синтаксический анализ использовался для выявления случаев, в которых формально корректные синтаксические решения приводили к снижению функциональной прозрачности текста. В ряде случаев привлекались элементы дискурсивного анализа, учитывающие жанровые и регистровые особенности документации.

Распределение выявленных переводческих неудач по типам осуществлялось в соответствии с авторской классификацией, представленной в теоретической части исследования. Каждая неудача относилась к одному из базовых типов либо, при наличии межуровневого взаимодействия, классифицировалась как гибридная. При дальнейшем анализе языкового материала учитывались как количественные параметры (частотность, распределение по документациям и разделам), так и качественные характеристики, отражающие специфику реализации переводческих неудач в различных жанровых зонах документации. Итоговое распределение переводческих неудач по типам представлено на Рисунке 7.

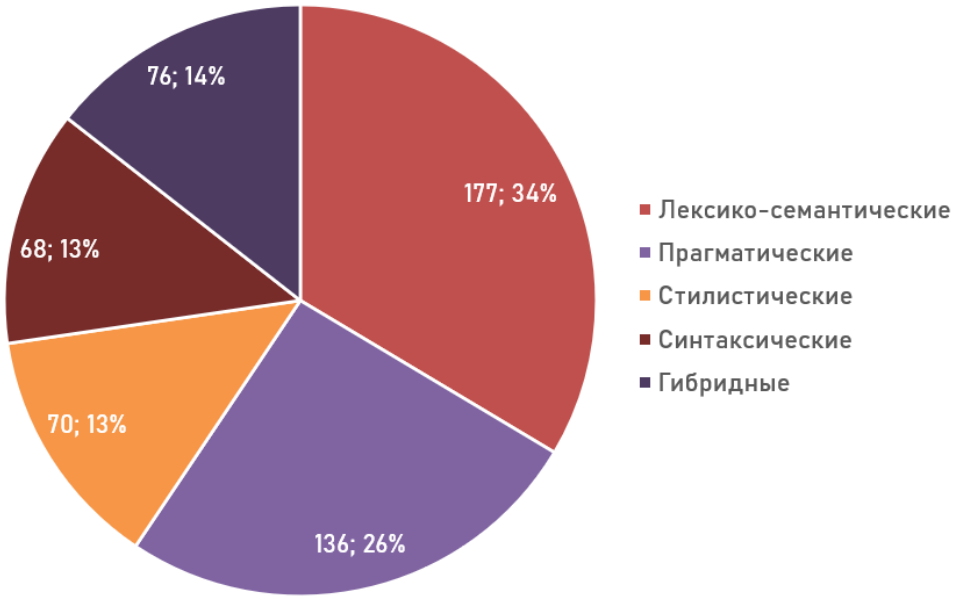


Рисунок 7. Распределение переводческих неудач по типам

3.2. Лексико-семантические переводческие неудачи

3.2.1. Статистическая характеристика и общая типология

В ходе анализа корпуса было выявлено 177 лексико-семантических переводческих неудач, что составляет около 33,59% от общего объёма материала (527 зафиксированных переводческих неудачи). Тем самым лексико-семантические неудачи образуют наиболее представительную группу в корпусе собранного языкового материала и играют ключевую роль в формировании общего коммуникативного эффекта перевода документации по веб-разработке.

Распределение лексико-семантических переводческих неудач по анализируемым документациям характеризуется выраженной асимметрией. Наибольшее их количество зафиксировано в документации React JS – 55,37% всех случаев данного типа (98 микроконтекстов). Меньшая доля приходится на MDN Web Docs (29,38%, 52 случая), и минимальное количество выявлено в документации Vue.js (15,25%, 27 случаев). При этом доля лексико-семантических неудач в документации React JS остаётся высокой даже с учётом её количественного преобладания в корпусе исследуемого языкового материала, что указывает на системный характер наблюдаемых сдвигов.

Анализ распределения лексико-семантических неудач по структурным зонам документации показывает, что они преимущественно локализуются в обучающих и

пояснительных разделах, где формируется базовая концептуальная картина технологии и активно используется специализированный метаязык. В документации React JS подобные неудачи особенно часто выявляются во вводных и направляющих микроконтекстах, задающих траекторию дальнейшего освоения материала. В документации MDN Web Docs и Vue.js лексико-семантические неудачи, как правило, имеют более локальный характер и реже образуют протяжённые серии.

Качественный анализ материала позволяет выделить несколько основных подгрупп лексико-семантических переводческих неудач.

Во-первых, это неудачи, связанные с переводом, сопровождающимся избыточной смысловой конкретизацией, при котором в переводе на русский язык актуализируются дополнительные смысловые компоненты, отсутствующие в оригинале, либо происходит смещение смыслового акцента микроконтекста.

Во-вторых, к данной группе относятся случаи сужения или расширения значения лексем, не приводящие к утрате предметного содержания, но влияющие на способ концептуализации описываемого явления.

В-третьих, отдельную и методологически значимую подгруппу образуют лексико-семантические неудачи, связанные с использованием терминологии. К ним относятся случаи неверного употребления терминов, несистематического использования терминологических единиц, а также варьирования нескольких переводных эквивалентов одного и того же термина в пределах одного микроконтекста или смежных фрагментов текста.

Следует подчеркнуть, что терминологические неудачи данного типа не обязательно сопровождаются искажением фактической информации. Однако их систематическое накопление приводит к нарушению целостности терминологической картины, что затрудняет формирование устойчивых понятийных связей у реципиента. В результате этого один и тот же концепт может интерпретироваться как несколько различных, что повышает риск неверной

интерпретации последующих микроконтекстов и снижает эффективность трансфера знаний.

Сопоставление документаций React JS, MDN Web Docs и Vue.js показывает, что в переводах MDN Web Docs и Vue.js терминологические решения в большей степени характеризуются стабильностью и воспроизводимостью, тогда как в документации React JS наблюдается более высокая вариативность терминологических соответствий. Это различие позволяет рассматривать терминологические лексико-семантические неудачи как один из факторов, объясняющих общее количественное преобладание неудач в переводе документации React JS.

3.2.2. Лексико-семантические переводческие неудачи при передаче метаязыка документации

Как показал анализ корпуса, значительная часть лексико-семантических переводческих неудач связана с передачей метаязыка документации, то есть лексем, описывающих структуру, этапность и логику освоения материала. В рамках корпуса языкового материала к данной группе отнесен 61 микроконтекст, из которых более 40 выявлены в документации React JS. Подобные неудачи локализуются преимущественно в вводных и направляющих фрагментах текста, где формируется представление о характере и назначении последующих разделов.

Рассмотрим микроконтекст (1) из документации React JS:

(1) This **tutorial** will give you a good starting point to build React applications.

Эта **практическая часть** даст вам хорошую отправную точку для создания приложений на React.

В данном микроконтексте ключевой метаязыковой лексемой является слово *tutorial*, которое в оригинале маркирует целостный обучающий маршрут, предполагающий поэтапное освоение материала. В переводе на русский язык данная лексема может передаваться как «практическая часть», что говорит о сужении смыслового объёма по сравнению с лексической единицей текста оригинала. В результате обучающий маршрут начинает интерпретироваться как вспомогательный или прикладной элемент, а не как структурно организованный курс. Подобный лексико-семантический сдвиг изменяет статус соответствующего раздела в общей логике документации.

Аналогичный механизм наблюдается в микроконтексте (2):

(2) The **setup** for this tutorial is simple and should take only a few minutes.

Подготовка к этой практической части проста и займёт всего несколько минут.

Здесь лексема *setup* в тексте оригинала обозначает отдельный обязательный этап, предшествующий основному обучающему процессу. В переводе на русский язык она передается как «подготовка», что само по себе допустимо, однако в сочетании со словосочетанием «практическая часть», служащим для передачи лексемы исходного текста *tutorial* формирует иную интерпретационную рамку. В результате микроконтекст текста перевода начинает восприниматься как описание вспомогательного действия перед прикладным упражнением, тогда как в оригинале речь идёт о структурном элементе обучающего маршрута. Таким образом, лексико-семантическая неудача носит кумулятивный характер и проявляется в серии взаимосвязанных микроконтекстов.

Рассмотрим ещё один микроконтекст:

(3) In this **guide**, you'll learn how React thinks about UI.

В этом **руководстве** вы узнаете, как React работает с пользовательским интерфейсом.

В оригинале лексема *guide* используется как элемент метаязыка, указывающий на объяснительный формат изложения и акцент на концептуальной модели технологии. В переводе на русский язык она передаётся как «руководство», что формально соответствует словарному значению, однако последующий перевод *thinks about UI* как «работает с пользовательским интерфейсом» приводит к смещению смыслового акцента с концептуального объяснения на описание функционального поведения. В результате микроконтекст оригинала (3) воспринимается скорее как часть технического описания, а не как объяснение принципов.

Сопоставление с документациями MDN Web Docs и Vue.js показывает, что подобные лексико-семантические сдвиги при передаче метаязыка в них фиксируются существенно реже. Даже в случаях использования оценочно или структурно нагруженных лексем перевод, как правило, сохраняет нейтральный характер и не изменяет функциональный статус соответствующего фрагмента текста. В документации React JS, напротив, интерпретирующий перевод метаязыка носит более системный характер и затрагивает повторяющиеся типы микроконтекстов, что усиливает его воздействие на восприятие документации в целом.

3.2.3. Лексико-семантические переводческие неудачи, связанные с терминологическим обеспечением согласованного употребления терминов

Лексико-семантические переводческие неудачи, связанные с обеспечением согласованного употребления терминов, образуют в корпусе языкового материала самостоятельный и методологически значимый подтип, поскольку затрагивают сам понятийный аппарат технологий. Неудач подобного подтипа выявлено существенно больше в документации к React JS, чем в документации к MDN Web Docs и Vue.js. Такая асимметрия согласуется с общей тенденцией: при меньшей последовательности переводческих решений терминологический слой становится зоной накопительного расхождения, влияющего на интерпретацию последующих фрагментов документации.

Характерной особенностью терминологических неудач является то, что они нередко проявляются не в одном изолированном микроконтексте, а в виде вариативности соответствий, когда одно и то же понятие в оригинале обозначается разными лексемами в переводе на русский язык.

Такая вариативность нарушает целостность терминологической картины и формирует у реципиента нестабильные понятийные связи, что может повышать вероятность неверной интерпретации следующих микроконтекстов, опирающихся на ранее введённые понятия.

Показательным случаем является использование лексемы *sections* в тексте оригинала, передаваемой в переводе вводной части документации к React JS как «часть» и как «раздел». Хотя формально лексемы *section* и «раздел» обладают высокой степенью семантической близости, одновременная фиксация двух переводческих соответствий приводит к терминологической неустойчивости в пределах даже одного микроконтекста. В результате реципиент получает два потенциальных обозначения для одного структурного элемента, что затрудняет соотнесение данного фрагмента с системой навигации документации. Подобная вариативность приобретает особую значимость в документации, в которой лексемы,

указывающие на разделы документации, регулярно повторяются и используются как ориентиры при восприятии материала.

Терминологическая нестабильность проявляется и в микроконтекстах, связанных с переводом устойчивых обозначений сред и режимов разработки. Рассмотрим следующий микроконтекст:

(4) There are two ways to complete this tutorial: you can either write the code in your browser, or you can set up **a local development environment** on your computer.

Есть два варианта прохождения практической части — вы можете писать код в браузере, либо настроить **окружение для разработки** на компьютере.

Терминологически значимым компонентом в микроконтексте оригинала является лексема *local*, уточняющая характер окружения. Речь здесь идёт о среде, развёрнутой локально (на стороне пользователя). В переводе на русский язык данный лексический компонент опускается, вследствие чего микроконтекст начинает восприниматься как общее указание на «настройку окружения», без маркирования локальности. При единичном прочтении это не препятствует пониманию, однако при повторяемости подобных опущений в серии микроконтекстов реципиент может не зафиксировать различие между локальными и иными режимами работы (например, браузерными, удалёнными, облачными). Таким образом, терминологическая редукция влияет на последующую интерпретацию инструктивных фрагментов, в которой *local* может выступать смысловым ограничителем применимости.

Помимо этого, встречаются случаи, когда термин сохраняется в виде англоязычной лексемы, не будучи включённым в систему русскоязычных соответствий. Примером служит микроконтекст (5):

(5) Create React App doesn’t handle **backend logic** or databases; it just creates a **frontend** build pipeline, so you can use it with any **backend** you want.

RU: Create React App не обрабатывает **бэкенд логику** или базы данных, он только предоставляет команды для сборки **фронтенда**, поэтому вы можете использовать его с любым **бэкэндо**м.

В данном случае фиксируется сразу два уровня терминологической неустойчивости. Во-первых, в переводе на русский язык наблюдается конкуренция вариантов транслитерации (backend → бэкенд / бэкэнд), что препятствует формированию устойчивой терминологической нормы в пределах одной и той же документации. Во-вторых, сочетание «бэкенд логика» демонстрирует недостаточную терминологическую согласованность: англоязычный термин не интегрируется в привычную русскоязычную модель (например, «логика бэкенда», «серверная логика»), что усложняет интерпретацию исходного микроконтекста для реципиента с различным уровнем профессиональной подготовки. Подобные случаи могут демонстрировать накопительный эффект. При вариативности транслитерации реципиент может не распознавать тождественность понятий в разных фрагментах текста и воспринимать их как разные понятия.

Терминологический подтип лексико-семантических неудач проявляется также в передаче отдельных общетехнических лексем, активно и многократно используемых в документации. В подобных случаях объектом анализа выступает не микроконтекст, а лексема как повторяющаяся единица, функционирующая в серии смежных фрагментов текста. Показательным в этом отношении является передача лексемы *websites*, которая в переводе на русский язык реализуется в нескольких вариантах: «сайты», «вебсайты» и «веб-сайты».

Наличие нескольких графических и словообразовательных вариантов одной и той же лексемы, не разграниченных функционально и не закреплённых за различными контекстами употребления, препятствует унификации терминологической формы в пределах документации. В результате реципиент сталкивается с вариативностью обозначения базового объекта, которая не

сигнализирует о различии понятий, однако создаёт впечатление отсутствия единого терминологического решения.

Несмотря на кажущуюся незначительность подобных расхождений их значение оказывается существенным. Документация по веб-разработке опирается на повторяемость лексем как на один из ключевых механизмов когнитивной ориентации реципиента. Устойчивое воспроизведение терминологических обозначений позволяет соотносить новые фрагменты текста с ранее усвоенными элементами. Когда же одно и то же обозначение систематически варьируется, терминологическая картина становится фрагментированной, а реципиент утрачивает стабильные ориентиры при чтении.

Особенно значимыми подобные терминологические неудачи становятся в сочетании с другими проявлениями нестабильности терминологического слоя, такими как конкуренция транслитерационных вариантов или параллельное использование русскоязычных и англоязычных форм. В совокупности эти явления усиливают риск неверной интерпретации смежных микроконтекстов и снижают функциональную прозрачность перевода документации как инструмента трансфера знаний.

Показательны и случаи, когда выбор переводного термина вносит смысловое ограничение, не совпадающее с концептуальной нагрузкой оригинала. Рассмотрим (6):

(6) React elements are **immutable**.

Элементы React **иммутабельны**.

Термин *immutable* в англоязычном техническом дискурсе является стандартным и широко употребительным. Он функционирует как элемент устойчивой и прозрачной терминологической системы, в рамках которой его значение поддерживается регулярной сочетаемостью (например, *immutable data, immutable*

objects), частотностью употребления и наличием многочисленных поясняющих контекстов. Для англоязычного реципиента данный термин, как правило, не требует дополнительного пояснения, поскольку он встроен в когнитивную модель программирования и соотносится с конкретным набором операций и состояний, прежде всего с невозможностью изменения уже созданных объектов.

В переводе на русский язык используется лексема «иммутабельны», которая в профессиональной среде действительно встречается, однако её употребление не является равно доступным для всех реципиентов. В отличие от англоязычного оригинала, данный термин не опирается на столь же прозрачную и устоявшуюся систему русскоязычных терминологических связей и может восприниматься как узкоспециализированное обозначение, не подкреплённое привычным терминологическим окружением. В результате часть реципиентов интерпретирует данный микроконтекст через «узнавание» англицизма, тогда как другая часть потребителей перевода вынуждена реконструировать его значение на основе контекста.

В документации обучающего типа подобная ситуация повышает риск неверной интерпретации. Реципиент может не соотнести термин «иммутабельность» с конкретными практическими последствиями, такими как невозможность изменения состояния объекта после его создания, и, следовательно, некорректно интерпретировать последующие микроконтексты, опирающиеся на данное понятие. Таким образом, терминологическая неудача проявляется здесь не столько в выборе не подходящего по формальным признакам эквивалента, сколько в снижении прозрачности понятия для части аудитории по сравнению с англоязычным оригиналом, где данное понятие представлено в более доступной и системно организованной форме.

Наконец, терминологические неудачи проявляются в документации React JS в микроконтекстах, где обозначения операций и действий в концептуальном описании

технологии передаются разными лексемами, создающими конкурирующие модели интерпретации. Рассмотрим следующий микроконтекст:

(7) In this section, we will learn how to make the Clock component truly reusable and **encapsulated**. It will set up its own timer and update itself every second. We can start by **encapsulating** how the clock looks […]

В этой главе мы узнаем, как **инкапсулировать** и обеспечить многократное использование компонента Clock. Компонент самостоятельно установит свой собственный таймер и будет обновляться раз в секунду. Для начала, **извлечём** компонент, показывающий время […]

В микроконтексте (7) оригинала прослеживается следующая логика изложения мысли: в рамках данного раздела предполагается сделать компонент Clock многократно используемым и инкапсулированным; далее уточняется, что компонент будет самостоятельно управлять своим таймером и обновляться каждую секунду, после чего авторы предлагают начать с инкапсуляции того, как выглядит данный компонент. Таким образом, глагол *encapsulate* последовательно используется для обозначения как общего концептуального результата, то есть инкапсуляции компонента, так и конкретного действия, направленного на обособление его визуального представления. Термин функционирует единообразно и поддерживает целостную концептуальную модель: реципиент воспринимает все упомянутые действия как разные проявления одного и того же принципа.

В переводе на русский язык данная логика терминологически нарушается. В первом предложении текста перевода используется глагол «инкапсулировать», который является общепринятым русскоязычным эквивалентом англоязычной глагольной форме *to make encapsulated* и адекватно отражает оригинал. Однако в заключительной части микроконтекста (7) перевод реализуется посредством использования глагола «извлечём» в синтаксической конструкции «Для начала,

извлечём компонент, показывающий время»), которая должна была бы соответствовать англоязычной структуре *We can start by encapsulating how the clock looks*. В результате в пределах одного микроконтекста один и тот же англоязычный термин получает два различных переводных соответствия – «инкапсулировать» и «извлечь».

Подобная вариативность не является нейтральной. В русскоязычном техническом дискурсе глагол «извлекать» обычно ассоциируется с операцией механического выделения или перемещения элемента, тогда как «инкапсулировать» имеет другое понятийное предназначение – обозначать сокрытие внутренней реализации и ограничение доступа к ней. Реципиент может интерпретировать данные действия как разные и не связанные между собой операции, тогда как в оригинале они являются частью одного и того же процесса.

Иная ситуация наблюдается в микроконтексте (8), где в переводе на русский язык, напротив, происходит нивелирование терминологического различия, присутствующего в англоязычном оригинале, за счёт использования одного и того же переводного соответствия для разных терминов.

(8) When using React, you generally don’t need to call addEventListener to add **listeners** to a DOM element after it is created. Instead, just provide a **listener** when the element is initially rendered. When you define a component using an ES6 class, a common pattern is for an **event handler** to be a method on the class. For example, this Toggle component renders a button that lets the user toggle between “ON” and “OFF” states […]

При использовании React обычно не нужно вызывать addEventListener, чтобы добавить **обработчики** в DOM-элемент после его создания. Вместо этого добавьте **обработчик** сразу после того, как элемент отрендерился. В компоненте, определённом с помощью ES6-класса, в качестве **обработчика события** обычно выступает один из методов класса. Например, этот компонент Toggle рендерит

кнопку, которая позволяет пользователю переключать состояния между «Включено» и «Выключено»: […]

В англоязычном микроконтексте последовательно используются два термина – *event listener* и *event handler*, каждый из которых занимает собственное место в концептуальной модели работы с событиями. В первом случае речь идёт о механизме подписки на событие на уровне DOM (*addEventListener* и соответствующий *listener*), во втором – о функции или методе, обрабатывающем событие в рамках компонентной модели React (*event handler* как метод класса).

В переводе на русский язык оба термина передаются единым обозначением – «обработчик события». Так, в первом предложении говорится о необходимости «добавить обработчики» к DOM-элементу, а в дальнейшем утверждается, что «в качестве обработчика события обычно выступает один из методов класса». В результате различие между *listener* как объектом подписки и *handler* как функцией обработки события лексически нивелируется.

Подобное терминологическое сближение не является нейтральным. В оригинале противопоставление *event listener* и *event handler* поддерживает важное концептуальное разграничение между низкоуровневой моделью работы с событиями в DOM и более абстрактной моделью обработки событий в React-компонентах.

В переводе на русский язык данное разграничение размывается. Использование одного и того же термина «обработчик события» для обозначения двух принципиально различных понятий может привести к формированию у реципиента ложной интерпретации, в рамках которой *event listener* (т.е. «слушатель события») и *event handler* (т.е. «обработчик событий») воспринимаются как тождественные понятия. Это, в свою очередь, затрудняет понимание того, почему в React не рекомендуется напрямую использовать addEventListener и каким образом React абстрагируется от низкоуровневого DOM.

Особенно проблематичным данный терминологический сдвиг становится в обучающем контексте. Реципиент, не имеющий устойчивого представления о различиях между моделями обработки событий, может не распознать принципиальное отличие между подпиской (букв. «прослушкой») на событие и его обработкой, а также некорректно интерпретировать дальнейшие пояснения, связанные с синтетическими событиями и управлением ими внутри React. Таким образом, терминологическая лексико-семантическая неудача в данном микроконтексте проявляется не столько в искажении фактической информации, сколько в сведении двух концептуально различных терминов к одному переводческому эквиваленту, что снижает прозрачность понятийной структуры оригинального текста и усложняет её усвоение реципиентом.

Таким образом, терминологические лексико-семантические переводческие неудачи оказывают непосредственное влияние на интерпретацию текстов документации, поскольку затрагивают системность понятийных соответствий. В документации React JS они проявляются как в форме вариативности переводных эквивалентов и конкуренции транслитерационных вариантов, так и в виде терминологической редукции и смешения русскоязычных и англоязычных обозначений. Накопление подобных неудач приводит к фрагментации терминологической картины и повышает вероятность неверной интерпретации последующих микроконтекстов, опирающихся на уже введённые понятия. В результате перевод сохраняет предметную информативность, однако утрачивает часть своей функциональной состоятельности как инструмента трансфера знаний.

3.3. Прагматические переводческие неудачи

3.3.1. Статистическая характеристика и общая типология

В ходе анализа корпуса языкового материала было выявлено 136 прагматических переводческих неудач, что составляет 25,81% от общего объёма материала (527 случаев). Тем самым прагматические неудачи образуют количественно значимую группу в исследовательском корпусе и характеризуются устойчивой повторяемостью в различных разделах документации по веб-разработке.

Распределение прагматических переводческих неудач по анализируемым документациям характеризуется выраженной асимметрией. Наибольшее их количество зафиксировано в документации React JS – 59,56% всех случаев данного типа (81 микроконтекст). Существенно меньшая доля приходится на MDN Web Docs (28,68%, 39 случаев), и минимальное количество выявлено в документации Vue.js (11,76%, 16 случаев). При этом высокая доля прагматических неудач в документации React JS сохраняется независимо от тематического распределения материала, что указывает на системный характер прагматических сдвигов в переводе.

Анализ распределения прагматических неудач по структурным зонам документации показывает, что они преимущественно локализуются в вводных, инструктивных и пояснительных фрагментах, ориентированных на управление действиями и ожиданиями реципиента. В документации React JS такие неудачи особенно часто выявляются в микроконтекстах, сопровождающих пошаговое изложение материала и объяснение принципов работы компонентов. В документациях MDN Web Docs и Vue.js прагматические неудачи, как правило, имеют более локальный характер и реже образуют протяжённые серии в пределах одного раздела.

Качественный анализ материала позволяет выделить несколько основных подтипов прагматических переводческих неудач.

Во-первых, это неудачи, связанные с изменениями модальной семантики, при которых в переводе на русский язык сглаживаются или, напротив, усиливаются значения возможности, вероятности и необходимости, заданные в оригинале. Во-вторых, к данной группе относятся прагматические неудачи, обусловленные искажением директивности и инструктивности, в результате чего меняется статус высказывания как рекомендации, совета или обязательного для выполнения указания. В-третьих, отдельную подгруппу образуют прагматические неудачи, связанные с утратой эмоционального тона, характерного для обучающей документации. В-четвёртых, выделяются прагматические неудачи, затрагивающие невербальные элементы документации, включая взаимодействие текста с кодом, заголовками, гиперссылками и визуальной структурой страницы.

Следует отметить, что прагматические неудачи данного типа не сопровождаются искажением предметного содержания микроконтекстов. Однако их систематическое накопление приводит к изменению коммуникативного эффекта перевода. Реципиент по-иному интерпретирует степень обязательности действий, характер рекомендаций и общую тональность взаимодействия с документацией. В результате этого нарушается управляемость сценария чтения и освоения материала, что снижает эффективность предпринимаемого путем перевода трансфера знаний.

Сопоставление документаций React JS, MDN Web Docs и Vue.js показывает, что прагматические неудачи в переводах MDN Web Docs и Vue.js в большей степени носят локальный характер и реже затрагивают ключевые элементы организации обучающего текста. В документации React JS, напротив, прагматические сдвиги проявляются более системно и затрагивают повторяющиеся типы микроконтекстов, что позволяет рассматривать их как устойчивую особенность переводной версии документации.

3.3.2. Прагматические неудачи, связанные с модальной семантикой

Прагматические неудачи, связанные с модальной семантикой, затрагивают способы выражения возможности, вероятности, допустимости и необходимости действий в документации по веб-разработке. В англоязычном оригинале данные значения реализуются с помощью модальных глаголов и конструкций, которые позволяют гибко управлять интерпретацией текста, задавать степень обязательности рекомендаций и формировать вариативные сценарии взаимодействия реципиента с материалом. Перевод подобных микроконтекстов требует сохранения прагматической настройки высказывания.

В переводе на русский язык значения вероятности и возможности могут ослабляться либо, напротив, усиливаться, что приводит к изменению коммуникативного эффекта высказывания. Такие сдвиги не затрагивают фактическую сторону излагаемого материала, однако влияют на то, как реципиент интерпретирует допустимость тех или иных действий и степень ответственности за выбор дальнейших шагов.

Нарушения в передаче модальной семантики особенно значимы для обучающей документации, где авторы стремятся не навязывать жёсткие алгоритмы, а предлагать гибкие пути освоения технологии. При утрате модальных различий перевод может формировать более жёсткий и линейный сценарий чтения, чем предусмотрено оригиналом, либо, напротив, снижать степень директивности там, где она функционально оправдана. В результате реципиент может неверно интерпретировать статус рекомендаций, воспринимая их как обязательные требования или, напротив, как факультативные замечания.

Рассмотрим (9), в котором прагматический сдвиг обусловлен заменой нормативной модальности на субъективно-оценочную:

(9) Conceptually, components are like JavaScript functions. They accept arbitrary inputs (called “props”) and return React elements describing what should appear on the screen.

Во многом компоненты ведут себя как обычные функции JavaScript. Они принимают произвольные входные данные (так называемые «пропсы») и возвращают React-элементы, описывающие, что мы хотим увидеть на экране.

В англоязычном оригинале ключевым элементом является модальный глагол *should* в конструкции *what should appear on the screen*. В данном микроконтексте *should* не выражает субъективного желания, а задаёт нормативно-функциональную установку. Компоненты React возвращают описание того результата, который должен быть отображён в соответствии с вычисленным состоянием и переданными входными данными. Тем самым подчёркивается детерминированный характер рендеринга и соответствие результата внутренней логике работы компонента.

В переводе на русский язык данная нормативная модальность передаётся формой «что мы хотим увидеть на экране», выражающую модальность желания. В результате происходит прагматический сдвиг: предписываемый и ожидаемый результат работы компонента начинает интерпретироваться как субъективное намерение разработчика или пользователя. Нормативный характер высказывания ослабляется, а коммуникативная установка оригинала смещается в сторону оценочно-личностной.

Подобное изменение влияет на интерпретацию микроконтекста. Реципиент может воспринимать возвращаемый результат компонента как реализацию произвольного замысла («то, что мы хотим»), что снижает прозрачность объяснения компонентов как чистых функций. В контексте обучающей документации это

особенно значимо, поскольку данный фрагмент используется для формирования базового представления о принципах работы React.

Схожие прагматические сдвиги фиксируются и в микроконтекстах, в которых модальная семантика оригинала сочетает обязательность действия с предостерегающим компонентом, направленным на управление вниманием реципиента.

(10) You **have to be careful** about the meaning of this in JSX callbacks. In JavaScript, class methods are not bound by default. If you forget to bind this.handleClick and pass it to onClick, this will be undefined when the function is actually called.

При обращении к this в JSX-колбэках **необходимо** учитывать, что методы класса в JavaScript по умолчанию не привязаны к контексту. Если вы забудете привязать метод this.handleClick и передать его в onClick, значение this будет undefined в момент вызова функции.

В англоязычном оригинале (10) прагматическая установка высказывания формируется за счёт сочетания конструкции *have to* и оценочно-предостерегающего компонента *be careful*. Конструкция *you have to* задаёт высокий уровень обязательности, указывая на необходимость учитывать специфику поведения this. При этом глагольная группа *be careful* выполняет функцию прагматического предостережения, акцентируя потенциальную проблемность ситуации и необходимость повышенного внимания со стороны реципиента.

В переводе на русский язык данная установка видоизменяется. Модальный компонент *have to* передаётся через конструкцию «необходимо», что в целом сохраняет значение обязательности. Однако предостерегающий элемент *be careful* передается посредством использования глагола «учитывать», что приводит к сглаживанию эмоционально-прагматического тона звучания оригинала. Лексема

«учитывать» задаёт нейтральную, описательную установку и ориентирует реципиента на констатацию условия, а не на источник ошибок.

В результате изменяется коммуникативный эффект микроконтекста. В оригинале реципиенту предлагается осознавать возможные последствия невнимательности, что подкрепляется последующим сложноподчиненным предложением с придаточной частью условия (*If you forget to bind… this will be undefined*). В переводе же данная логика воспринимается более формально. Внимание реципиента переключается с предупреждения на перечисление фактов, что снижает степень вовлечённости кого во что??? и ослабляет превентивную функцию высказывания.

Наряду с примерами из документации React JS прагматические неудачи, связанные с модальной семантикой, выявляются и в переводах документации MDN Web Docs. Рассмотрим микроконтекст (11) из раздела, посвящённого HTML, в котором модальная установка текста оригинала связана с нормативными требованиями к описанию изображений на веб-странице.

(11) When using the alt attribute, you **should provide** meaningful alternative text that describes the content of the image for users who cannot see it.

При использовании атрибута alt **указывается** альтернативный текст, описывающий содержимое изображения.

Атрибут *alt* в HTML используется для текстового описания изображения и предназначен прежде всего для пользователей, которые не могут воспринимать визуальный контент, например, при использовании программ экранного доступа. Таким образом, его корректное заполнение имеет не столько техническую, сколько функциональную значимость в контексте доступности веб-контента.

В англоязычном оригинале использование модального глагола *should* является прагматически обоснованным. В данном микроконтексте *should* выражает

нормативную рекомендацию, указывая на ожидаемое и социально значимое действие разработчика. Формулировка не носит характера жёсткого предписания, однако подчёркивает, что добавление осмысленного альтернативного текста является правильной и предпочтительной практикой при использовании атрибута *alt*. В переводе на русский язык модальный компонент опущен. В переводе наблюдается безличная описательная конструкция («указывается альтернативный текст»), фиксирующая сам факт наличия атрибута и его функции, но не маркирующая степень нормативности данного действия. В результате прагматическая установка оригинала ослабляется: рекомендация, ориентированная на ответственное использование атрибута alt, воспринимается как нейтральное описание технической возможности.

Подобное смещение влияет на интерпретацию микроконтекста (11). Реципиент, не обладающий специализированными знаниями в области веб-разработки, может не распознать, что речь идёт о важном требовании, связанном с доступностью контента. Таким образом, прагматическая неудача в данном случае проявляется в утрате нормативной модальности, что снижает коммуникативную значимость высказывания и может способствовать восприятию описываемого действия как факультативного.

Наряду с рассмотренными микроконтекстами из документаций React JS и MDN Web Docs аналогичные прагматические сдвиги, связанные с модальной семантикой, выявляются и в переводе документации Vue.js. Приведем микроконтекст (12), в котором нормативная модальность оригинала нивелируется в переводе:

(12) In Vue, a component’s data option **must** be a function that returns a fresh object for each component instance.

В Vue параметр data компонента **является** функцией, которая возвращает объект данных для экземпляра компонента.

Для понимания данного фрагмента текста отметим, что параметр *data* в компоненте Vue определяет исходное состояние компонента. Требование возвращать новый объект для каждого экземпляра связано с предотвращением совместного использования состояния между разными экземплярами компонента. Тем самым формируется обязательное правило корректной работы приложения.

В англоязычном оригинале микроконтекста (12) модальный глагол *must* выполняет ключевую прагматическую функцию: он маркирует строгое нормативное требование, нарушение которого приводит к некорректному поведению компонентов. Представленная формулировка не оставляет реципиенту пространства для интерпретации и задаёт однозначную установку на обязательность описываемого решения. В переводе на русский язык данная нормативная модальность утрачивается. Конструкция *must be* передаётся на русский язык посредством нейтрального глагола бытия «является», в результате чего высказывание приобретает характер констатации факта, а не предписания. Русскоязычный реципиент может интерпретировать данный фрагмент как описание типичной реализации параметра *data*, а не как обязательное правило, от соблюдения которого зависит корректность работы компонентов.

Подобное изменение влияет на прагматический эффект микроконтекста (12). Для реципиента, не обладающего глубокими знаниями в области веб-разработки, различие между обязательным требованием и описательной характеристикой может оказаться неочевидным. В результате утрачивается сигнал о критичности данного условия, что повышает риск неверной интерпретации последующих пояснений и некорректного применения описываемого принципа на практике.

3.3.3. Прагматические неудачи, обусловленные искажением директивности и инструктивности

Прагматические неудачи, обусловленные искажением директивности и инструктивности, затрагивают способы управления действиями реципиента в тексте документации по веб-разработке. В англоязычном оригинале данные функции реализуются через систему императивных конструкций, которые формируют пошаговый сценарий освоения материала. Перевод подобных фрагментов предполагает сохранение коммуникативного статуса высказывания.

В переводе на русский язык в ряде случаев инструктивные формулы утрачивают побудительный характер и переводятся как нейтральные описательные конструкции, в других рекомендации или пояснения приобретают более жёсткий директивный статус, вовсе не предусмотренный оригиналом. Подобные изменения не приводят к искажению предметного содержания микроконтекстов, однако существенно влияют на интерпретацию реципиентом характера предлагаемых действий.

Особую значимость подобные неудачи приобретают в обучающих и вводных разделах документации, где управление последовательностью действий и распределением внимания реципиента является ключевым элементом коммуникативной организации текста. При искажении директивности нарушается логика пошагового изложения, а реципиент может неверно интерпретировать, какие действия являются обязательными, какие рекомендуемыми, а какие выполняют вспомогательную пояснительную функцию.

Следует отметить, что прагматические неудачи, связанные с ослаблением инструктивности за счёт перехода от императивных форм к описательным конструкциям, в анализируемом корпусе выявлены исключительно в документации React JS. В переводах MDN Web Docs и Vue.js подобные смещения императивных

указаний не зафиксированы, что позволяет рассматривать их как специфическую особенность перевода документации React.

Приведем в целях наглядности микроконтекст (13).

(13) **Do not modify** state directly. **Always use** setState or the state updater function returned by useState.

Состояние компонента напрямую **не изменяется**. Для обновления состояния **используется** setState или функция-обновитель, возвращаемая useState.

В англоязычном оригинале микроконтекста (13) представлена строгая по прагматике инструкция, оформленная посредством императива с отрицанием (*Do not modify*) и усиленную наречием *always*, которое дополнительно маркирует обязательность последующего действия. Коммуникативная установка высказывания оригинала однозначна: реципиенту прямо запрещается определённое действие и одновременно предлагается единственно допустимый способ работы с состоянием компонента. Подобная формулировка не оставляет пространства для альтернативной интерпретации и чётко фиксирует нормативное правило работы с состоянием в React. В переводе на русский язык данная директивная структура подвергается изменению. Императивная форма передаётся безличным описательным высказыванием («состояние компонента напрямую не изменяется»), которое по своему прагматическому статусу воспринимается как констатация факта, а не как запрет. Аналогичным образом вторая часть инструкции утрачивает в переводе характер прямого указания. Конструкция *Always use* передаётся в переводе посредством модально нейтрального глагола «используется», что переводит инструкцию в плоскость описания типичной практики.

В результате меняется коммуникативный эффект высказывания, приведенного в оригинале микроконтекста (13). Русскоговорящий реципиент может интерпретировать данный фрагмент не как обязательное правило, нарушение

которого приводит к некорректному поведению приложения, а как общее описание рекомендуемого подхода. Для пользователя перевода, не обладающего глубокими знаниями в области веб-разработки, различие между нормативным запретом и описанием «того, как обычно делается», может оказаться неочевидным.

В целом ряде микроконтекстов искажение директивности проявляется в переводе не посредством полной утраты инструкции или её усиление, а путем смягчения запретительной установки, присутствующей в оригинале.

Приведем лишь один пример.

(14) **Don’t use** array indexes as keys if the order of items may change. This can lead to performance issues and incorrect component state.

Не рекомендуется использовать индексы массива в качестве ключей, если порядок элементов может изменяться, так как это может привести к проблемам с производительностью и состоянием компонентов.

В англоязычном оригинале данный фрагмент оформлен как прямой запрет, выраженный через отрицательную императивную конструкцию *Don’t use*. Такая формулировка чётко маркирует недопустимость описываемого действия при выполнении указанного условия (*if the order of items may change*). Последующее предложение указывает на возможные негативные последствия нарушения запрета, что усиливает директивный эффект высказывания. В переводе на русский язык запретительная установка ослабляется. Императивная форма передаётся конструкцией «не рекомендуется использовать», которая по своему прагматическому статусу относится к сфере советов и предпочтительных практик. В результате запрет становится рекомендацией, а степень обязательности действия существенно снижается.

Подобное смягчение влияет на интерпретацию перевода в микроконтексте (14). Реципиент может воспринимать использование индексов массива в качестве ключей

не как принципиально некорректное решение в определённых условиях, а как нежелательный, но все же допустимый вариант.

Если в предыдущих микроконтекстах директивная установка оригинала в переводе ослаблялась, то в следующем фрагменте текста перевод усиливает инструктивность высказывания, придавая ему форму прямого указания, отсутствующую в оригинале.

(15) **We gave this ** a unique id HTML attribute. This will allow us to find it from the JavaScript code later and display a React component inside of it.

Затем **назначьте созданному ** уникальный атрибут id. Это позволит впоследствии найти тег из JavaScript-кода и отобразить React-компоненты внутри него.

В англоязычном оригинале данный фрагмент оформлен как описание уже выполненного действия. Использование формы прошедшего времени (*We gave this *…) фиксирует текущее состояние разметки и выполняет пояснительную функцию: авторы объясняют, почему элементу был присвоен атрибут *id*, и как это будет использоваться далее. Коммуникативная установка оригинала микроконтекста (15) носит разъяснительный характер и не содержит прямого побуждения к действию. В переводе на русский язык данная установка не прослеживается. Описательная конструкция передаётся императивной формой («назначьте»), в результате чего микроконтекст приобретает характер прямой инструкции. Реципиент воспринимает высказывание как указание к выполнению конкретного шага в действиях. Таким образом, в переводе происходит сдвиг от объяснения к директиве.

Подобное усиление директивности влияет на интерпретацию микроконтекста (15). Реципиент может воспринять данный фрагмент как обязательный шаг в последовательности действий, даже если в оригинале он выполняет

вспомогательную функцию и служит лишь для пояснения логики дальнейших операций.

3.3.4. Прагматические неудачи, связанные с утратой эмоционального тона

Одной из характерных особенностей документации по веб-разработке является активное использование эмоционально-прагматических средств и формирование комфортного режима взаимодействия с материалом. В англоязычных оригиналах такие средства особенно заметны в обучающих и вводных разделах, где авторы стремятся поддержать реципиента, обозначить допустимость ошибок и задать доброжелательный тон изложения. Эмоциональный тон в документации выполняет важную коммуникативную функцию. Он способствует снижению тревожности при освоении новой технологии, формирует ощущение диалога между автором и читателем и помогает реципиенту воспринимать материал как последовательный и доступный. Такие элементы, как ободряющие формулы, смягчающие комментарии и обращения к реципиенту, позволяют выстраивать обучающий сценарий, в котором техническая сложность компенсируется прагматической поддержкой.

В переводе на русский язык данные элементы нередко подвергаются нейтрализации. Эмоционально окрашенные или диалогически ориентированные высказывания передаются формально-описательными конструкциями, что приводит к утрате коммуникативного эффекта, заложенного в оригинале. При этом предметное содержание микроконтекстов, как правило, сохраняется, однако меняется характер взаимодействия с реципиентом текста перевода: предлагаемый в результате перевода текст приобретает более сухой и отстранённый характер.

Подобные прагматические сдвиги особенно заметны в тех фрагментах документации, которые ориентированы на сопровождение реципиента в процессе обучения. Утрата эмоционального тона не искажает фактическую информацию, но влияет на восприятие материала в целом, снижая его обучающий потенциал и ослабляя тем самым поддерживающую функцию текста.

Потеря эмоционального тона особенно наглядно проявляется в микроконтекстах, в которых оригинальный текст использует минимальные, но функционально значимые элементы оценки, направленные на формирование позитивного восприятия технологии. Подобные случаи в анализируемом корпусе систематически фиксируются именно в документации React JS, тогда как в переводах MDN Web Docs и Vue.js эмоционально-прагматические маркеры, как правило, сохраняются.

(16) React lets you build user interfaces out of individual pieces called components. **It’s fun!**

React позволяет создавать пользовательские интерфейсы из отдельных частей, называемых компонентами.

Вторая фраза англоязычного оригинала (*It’s fun!*) не несёт никакой дополнительной предметной информации, однако выполняет чётко выраженную прагматическую функцию. Краткая оценочная реплика формирует позитивный эмоциональный фон, снижает дистанцию между автором и реципиентом и задаёт установку на восприятие технологии как приятной в освоении. Подобные элементы характерны для вводных фрагментов документации React и направлены на вовлечение реципиента в дальнейшее чтение. В переводе на русский язык данный элемент не отражён. В результате микроконтекст (16) сводится к нейтральному описанию функциональных возможностей фреймворка. И хотя предметное

содержание исходного высказывания сохраняется, утрачивается эмоционально-прагматический компонент, задающий тон взаимодействия с реципиентом.

Систематичность подобных изменений в документации React позволяет предположить, что переводчик сознательно устраняет оценочные элементы как «избыточные» с точки зрения передачи технического содержания. Однако в условиях обучающей документации такие элементы выполняют коммуникативно значимую функцию, влияя на мотивацию и вовлечённость реципиента в понимание сообщаемого.

В следующем микроконтексте (17) утрата эмоционального тона в переводе проявляется в отсутствии элемента, который должен был смягчать тревожность реципиента при появлении предупреждений.

(17) **Don’t worry!** If you see this warning, it usually means that React is helping you catch a potential problem early.

Если вы видите это предупреждение, как правило, это означает, что React помогает выявить потенциальную проблему на раннем этапе.

В англоязычном оригинале вводная фраза *Don’t worry!* задаёт интерпретационную рамку для последующего пояснения. Предупреждение заранее представляется как полезный сигнал, свидетельствующий о корректной работе механизмов фреймворка. Данный эмоциональный маркер используется для того, чтобы скорректировать ожидаемую реакцию реципиента ещё до восприятия основной информации. В переводе на русский язык данный элемент отсутствует. Перевод микроконтекста (17) начинается непосредственно с описательного пояснения, в результате чего не отмечается смягчающий эффект, присутствующий в оригинале. Высказывание приобретает нейтральный, отстранённый характер.

При отсутствии такого элемента реципиент может воспринимать предупреждение как указание на ошибку или некорректное поведение кода, особенно

в случае недостаточного опыта работы с React. В оригинале такая интерпретация блокируется уже на уровне формулировки, тогда как в переводе она представляется возможной.

Прагматическая неудача в данном микроконтексте проявляется в утрате эмоционального маркера, который в оригинале снижает тревожность и помогает сформировать более спокойное отношение к диагностическим сообщениям. При сохранении информационного содержания меняется общий тон высказывания, что в совокупности с аналогичными случаями способствует формализации перевода и ослаблению поддерживающего характера документации.

3.3.5. Прагматические неудачи, затрагивающие невербальные элементы документации

Наряду с вербальными компонентами текста, документация по веб-разработке опирается на широкий спектр невербальных элементов, участвующих в формировании общего коммуникативного эффекта. К таким элементам относятся примеры кода, заголовки, гиперссылки, визуальные акценты, а также пространственная организация материала на странице. В совокупности они образуют единый прагматический контур, влияющий на интерпретацию текста и действия реципиента.

Прагматические неудачи данного подтипа возникают в тех случаях, когда перевод вербальной части документации осуществляется без учёта её взаимодействия с невербальным окружением. При этом предметное содержание отдельных микроконтекстов может сохраняться, однако нарушается согласованность

между текстом и сопровождающими его элементами. В результате реципиент сталкивается с рассинхронизацией между тем, что говорится в тексте, и тем, что демонстрируется или предполагается на уровне структуры страницы, кода или навигации.

Особую значимость подобные неудачи приобретают в обучающих и инструктивных разделах, где невербальные элементы выполняют направляющую и уточняющую роль. Примеры кода, визуально выделенные фрагменты и гиперссылки часто служат опорными точками для понимания материала и задают порядок его освоения. При нарушении их согласованности с текстом перевод утрачивает часть своей прагматической эффективности, даже если отдельные высказывания остаются формально корректными.

Прагматические неудачи, затрагивающие невербальные элементы документации, проявляются по-разному в каждой из анализируемых документаций, что обусловлено спецификой их структуры, принципами навигации и характером взаимодействия с реципиентом. В документации MDN Web Docs они затрагивают прежде всего интерфейсные и навигационные элементы.

Рассмотрим соответствующий пример на материале русскоязычной версии MDN Web Docs (раздел, посвящённый разметке HTML). Анализ визуальной структуры страницы показывает, что значительная часть элементов интерфейса сохраняется в англоязычном виде (см. Рисунок 8). К таким элементам относятся верхнее навигационное меню, выпадающие списки разделов, элементы фильтрации и часть вспомогательных подписей. Для реципиента, не владеющего английским языком в достаточной степени, подобная смешанная языковая среда может быть затруднительной для восприятия и ориентации в материале.

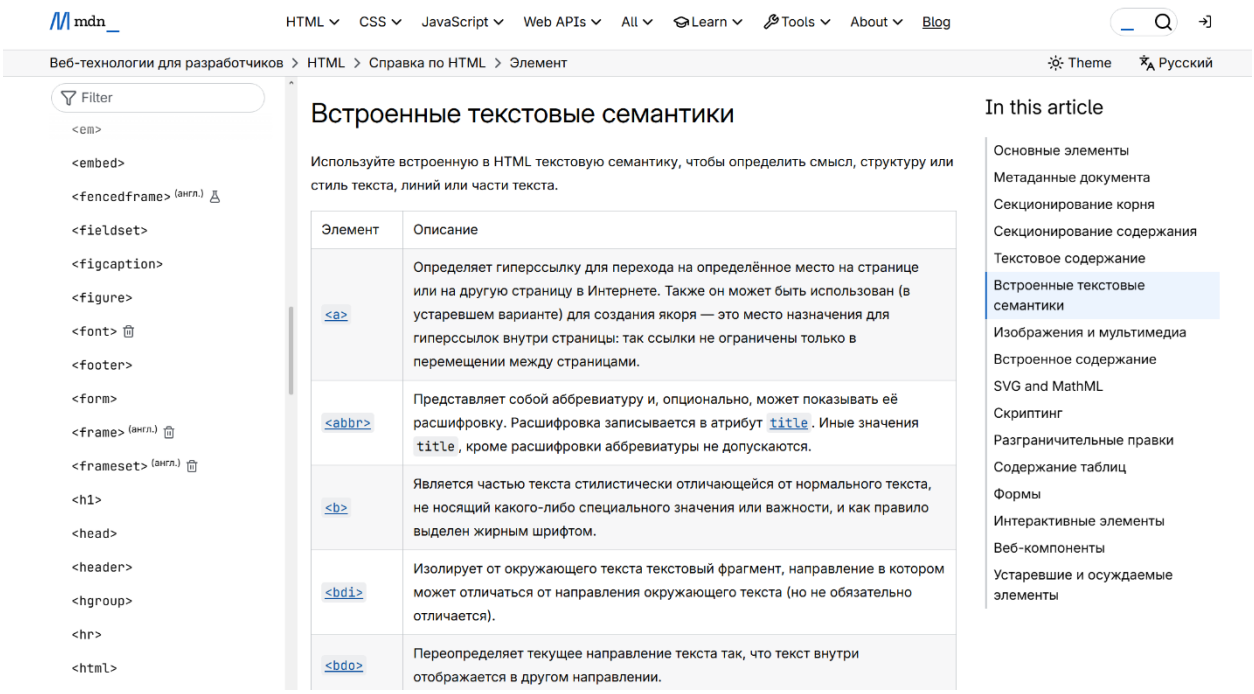


Рисунок 8. Внешний вид интерфейса MDN Web Docs

При этом следует отметить, что наличие непереведённых фрагментов в интерфейсе не всегда свидетельствует о переводческой неудаче. В ряде случаев подобные элементы относятся к недавно обновлённым разделам или функциональным компонентам сайта, которые на момент анализа могли не быть переведены по объективным причинам. Такие фрагменты, как правило, не несут ключевой коммуникативной нагрузки и не препятствуют пониманию основного содержания статьи. Однако в рассматриваемом примере непереданными оказываются прагматически значимые элементы, напрямую влияющие на взаимодействие реципиента с документацией: названия разделов в навигационной панели, элементы структурирования материала, а также интерфейсные компоненты, ориентированные на управление чтением и выбор траектории работы с текстом. Именно эти элементы задают сценарий взаимодействия реципиента с документацией и выполняют направляющую функцию.

В результате возникает прагматический разрыв между вербальным содержанием статьи, представленным на русском языке, и невербальной оболочкой страницы, частично сохраняющей англоязычную форму. Реципиент вынужден одновременно интерпретировать текст и ориентироваться в интерфейсе на другом языке, что повышает когнитивную нагрузку и снижает связность восприятия документации как единого обучающего ресурса. Подобная рассинхронизация особенно ощутима для пользователей без профессионального опыта работы с англоязычной документацией. В этом случае непереведённые элементы навигации и взаимодействия с реципиентом перестают выполнять свою поддерживающую роль и требуют дополнительного усилия для интерпретации, несмотря на то что основное текстовое содержание статьи формально доступно.

Отдельного рассмотрения в рамках прагматических неудач, затрагивающих невербальные элементы документации, заслуживают интерактивные примеры кода, активно используемые в MDN Web Docs. Подобные примеры представляют собой элементы взаимодействия, предполагающие запуск кода, изменение параметров и немедленное наблюдение результата. Тем самым они выполняют обучающую функцию, направляя действия реципиента.

Рассмотрим пример (см. Рисунок 9). В русскоязычной версии документации интерактивный пример формально интегрирован в текст, однако его содержательная часть полностью воспроизводит англоязычный вариант без адаптации для русскоязычного реципиента. Это проявляется, в частности, в использовании англоязычных имён и комментариев внутри кода, а также в сохранении англоязычной логики демонстрации результата.

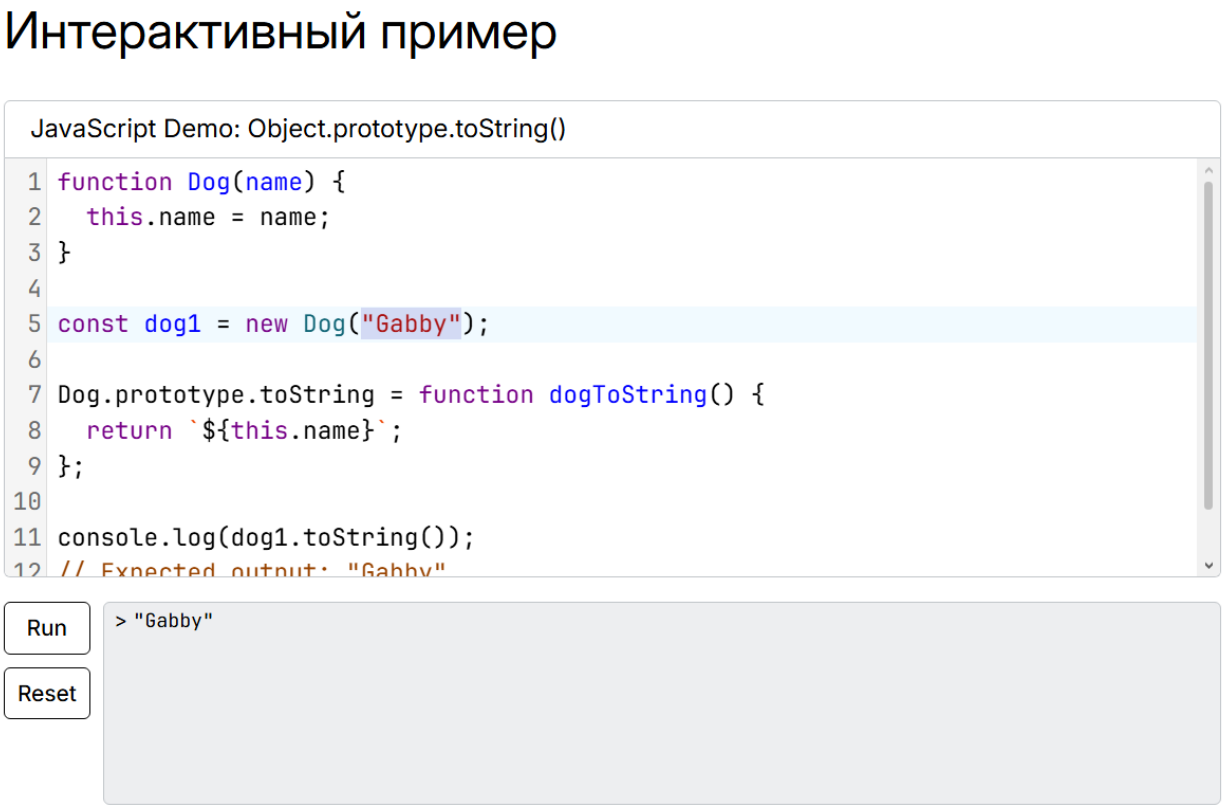


Рисунок 9. Интерактивный пример в русскоязычной версии MDN Web Docs

Подобное решение нельзя рассматривать как чисто техническое. Интерактивный пример ориентирован на непосредственное взаимодействие реципиента с материалом и, в отличие от статичного текста, активирует дополнительный прагматический уровень восприятия. Использование, например, имени Gabby (на латинице) не препятствует пониманию работы кода, однако создаёт дополнительный языковой барьер для реципиента, не владеющего английским языком. В условиях русскоязычной документации подобные элементы могли бы быть адаптированы, например, за счёт использования нейтральных или русскоязычных имён, не влияющих на технический смысл примера.

Отсутствие такой адаптации приводит к прагматическому рассогласованию: текстовое объяснение ориентировано на русскоязычного реципиента, тогда как интерактивный компонент продолжает функционировать в англоязычной парадигме.

В результате реципиент вынужден постоянно переключаться между языковыми кодами в пределах одного обучающего сценария, что повышает когнитивную нагрузку и снижает шансы на целостное восприятие материала.

В документации React JS прагматические неудачи, затрагивающие невербальные элементы, имеют несколько иную конфигурацию по сравнению с MDN Web Docs. В анализируемом корпусе не выявлены случаи рассинхронизации между языком вербального текста и языком интерфейса. Элементы навигации, управляющие компоненты страницы и сопровождающие подписи в русскоязычной версии документации React последовательно переведены и образуют единое коммуникативное пространство (см. Рисунок 10). Аналогичным образом примеры кода в целом адаптированы для русскоязычного реципиента и не содержат элементов, препятствующих восприятию материала.

В то же время в документации React фиксируется иной тип прагматической неудачи, связанный с функционированием гиперссылок. В ряде случаев русскоязычный текст содержит ссылки, ведущие на англоязычные внешние ресурсы или на разделы оригинальной документации, не имеющие перевода. Формально подобные ссылки сохраняют свою информативную ценность и обеспечивают доступ к дополнительным материалам, однако с прагматической точки зрения они нарушают целостность обучающего сценария.

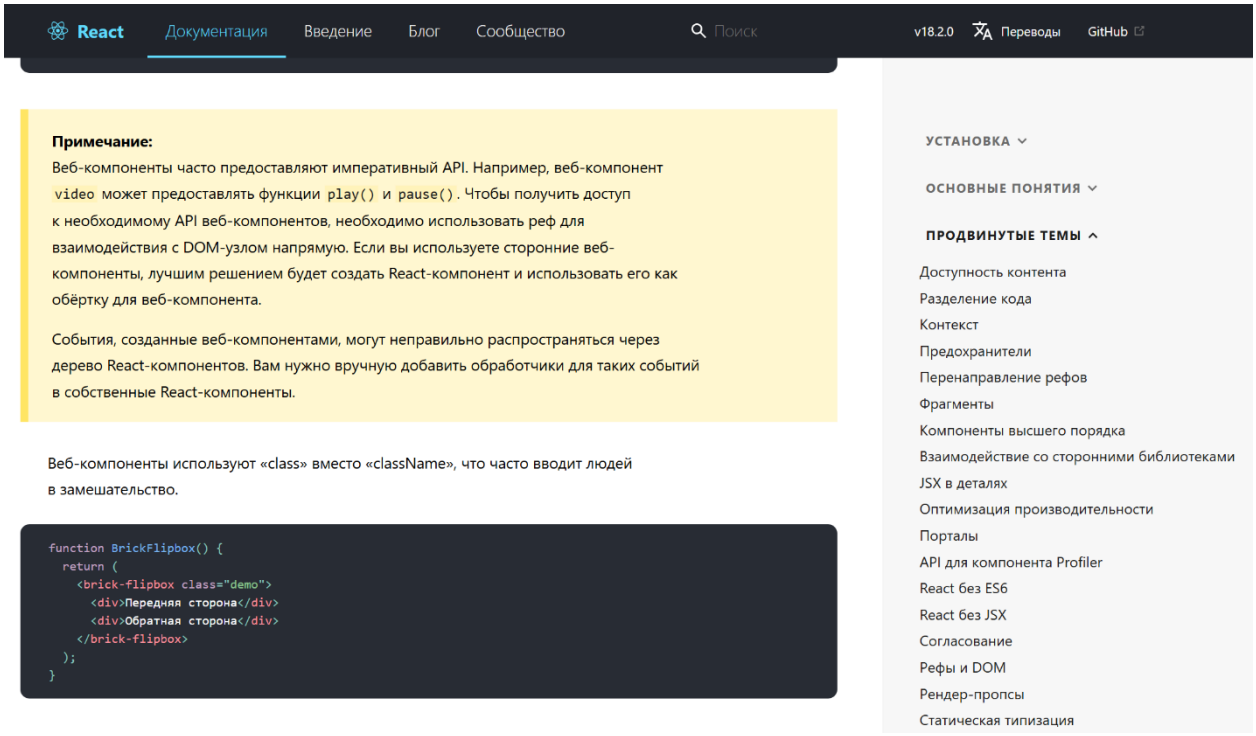


Рисунок 105. Внешний вид интерфейса документации к React JS

Для русскоязычного реципиента гиперссылка выполняет навигационную функцию: она предполагает логичное продолжение чтения в том же коммуникативном режиме, что и основной текст. При переходе по ссылке на англоязычный ресурс реципиент оказывается в иной языковой и прагматической среде, что требует дополнительного усилия для интерпретации информации. В результате нарушается ожидаемая непрерывность взаимодействия с документацией.

Подобная ситуация может быть объяснена объективными причинами, связанными с отсутствием перевода отдельных материалов или с ориентацией React на глобальное сообщество разработчиков. Однако с точки зрения прагматического анализа перевода данное решение приводит к частичному смещению коммуникативной направленности: основной текст ориентирован на русскоязычного реципиента, тогда как часть навигационных траекторий предполагает владение английским языком.

Документация Vue.js в русскоязычной версии демонстрирует наиболее последовательный и прагматически выверенный подход к переводу невербальных

элементов среди всех анализируемых ресурсов. В ходе анализа не выявлено системных неудач, связанных с рассинхронизацией вербального текста и интерфейсной оболочки: элементы навигации, заголовки разделов, управляющие кнопки и вспомогательные подсказки последовательно переведены на русский язык и образуют целостную коммуникативную среду.

Примеры кода в документации Vue.js также адаптированы для русскоязычного реципиента. Комментарии и поясняющие подписи не создают дополнительных языковых барьеров и не требуют постоянного переключения между языками при работе с материалом. Это способствует сохранению непрерывности обучающего сценария и снижает когнитивную нагрузку при восприятии документации.

Отдельного внимания заслуживает работа с гиперссылками. В отличие от документации React JS, ссылки, ведущие на внешние англоязычные ресурсы, в документации Vue.js, как правило, явно маркируются. Так, в интерфейсе указывается, что переход осуществляется на англоязычный ресурс, например, посредством пометы «(на англ.)» (см. Рисунок 11). Подобная маркировка позволяет реципиенту заранее скорректировать ожидания и осознанно принять решение о переходе по ссылке, не нарушая логики взаимодействия с документацией.

Следует отметить, что в исследовательском корпусе зафиксированы единичные случаи, в которых подобная маркировка отсутствует. Однако данные примеры носят несистемный характер и не формируют устойчивого паттерна прагматических неудач. В целом подход, реализованный в документации Vue.js, свидетельствует о внимании переводчиков к прагматической организации интерфейса и навигационных элементов.

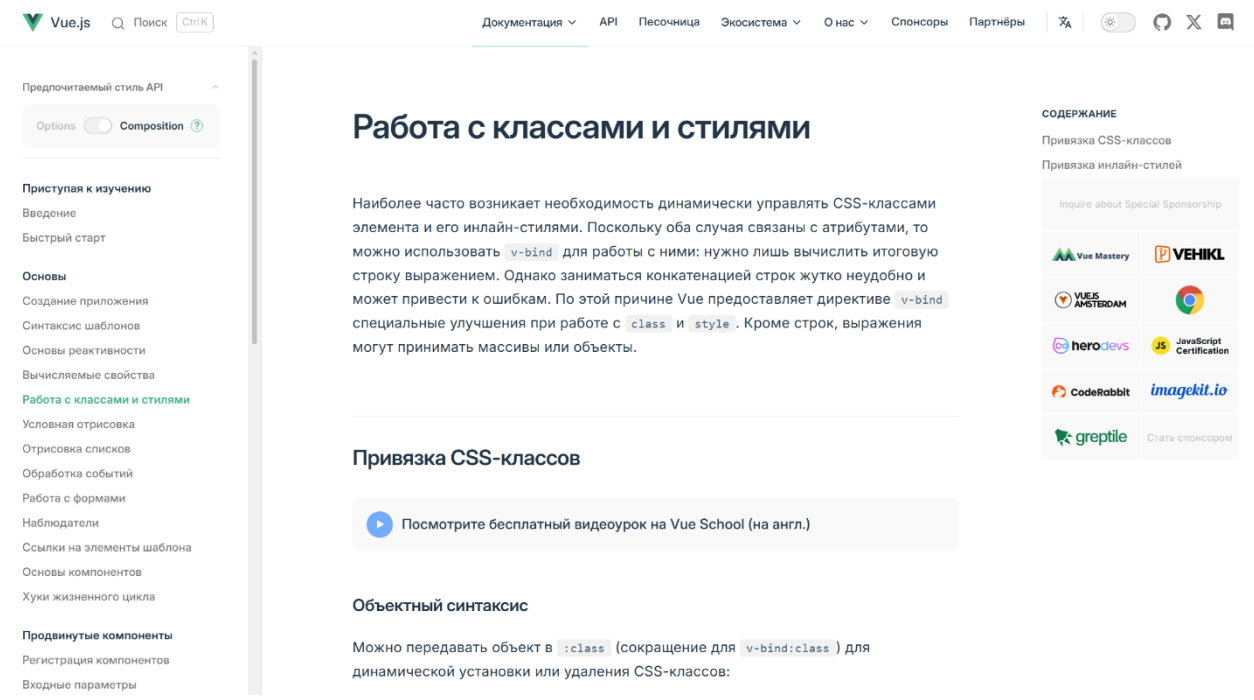


Рисунок 11. Внешний вид интерфейса документации к Vue.js

Таким образом, анализ прагматических переводческих неудач показал, что при сохранении предметного содержания оригинального текста в переводе систематически изменяется характер взаимодействия реципиента с документацией. Прагматические сдвиги затрагивают способы управления интерпретацией, ожиданиями и действиями реципиента и проявляются на разных уровнях организации документационного текста. Наиболее выраженный и системный характер прагматические неудачи имеют в документации React JS, где они затрагивают модальную семантику, директивность и эмоциональный тон высказываний, а также отдельные элементы навигационного сценария. В документации MDN Web Docs прагматические неудачи носят преимущественно локальный характер и связаны главным образом с рассинхронизацией вербального текста и невербальных компонентов. Документация Vue.js демонстрирует наиболее последовательный подход к прагматической адаптации и характеризуется минимальным количеством неудач данного типа.

В целом прагматические переводческие неудачи не приводят к искажению фактической информации, однако снижают управляемость обучающего сценария и ослабляют коммуникативную эффективность перевода. Это позволяет рассматривать прагматический уровень как одну из ключевых зон риска при переводе и локализации документации по веб-разработке.

3.4. Стилистические переводческие неудачи

3.4.1. Статистическая характеристика и общая типология

В ходе анализа корпуса было выявлено 70 стилистические переводческие неудачи, что составляет 13,28% от общего объёма материала (527 случаев). Тем самым стилистические неудачи образуют одну из наименее многочисленных групп в корпусе. Их рассмотрение в настоящей работе осуществляется в русле стилистики дискурса, поскольку анализируемые случаи затрагивают жанрово-регистровую организацию текста, его адресатную направленность и способ функционирования в профессиональной коммуникации.

Распределение стилистических переводческих неудач в переводе текстов документации по веб-разработке характеризуется выраженной неравномерностью. Наибольшее количество случаев с неудачами стилистического характера зафиксировано в документации React JS – 34, что составляет 48,57% всех неудач данного типа во всех исследуемых корпусах. Меньшая доля приходится на MDN Web

Docs – 28 случаев (40,00%), и минимальное количество стилистических неудач выявлено в документации Vue.js – 8 случаев (11,43%).

Анализ распределения стилистических неудач по структурным зонам документации показывает, что они преимущественно локализуются в заголовках, вводных пояснительных микроконтекстах, описаниях принципов работы технологии, а также во фрагментах, сочетающих техническое объяснение с элементами диалогичности. Иными словами, стилистические сдвиги, то есть изменения в тональности, уровне формальности и манере изложения текста, возникают прежде всего там, где переводчику необходимо удерживать баланс между профессиональной точностью и доступностью изложения.

В документации React JS подобные неудачи особенно часто выявляются в обучающих разделах, где оригинал сочетает технический стиль с умеренно разговорной, вовлекающей подачей.

В документации MDN Web Docs стилистические неудачи, как правило, распределены более точечно и чаще всего касаются отдельных заголовков или неудачно выбранных формулировок, не складывающихся в устойчивую тенденцию.

В документации Vue.js такие неудачи встречаются реже всего и в целом не нарушают общей стилистической цельности перевода.

Качественный анализ материала позволяет выделить два противоположных друг другу подтипа стилистических переводческих неудач.

Во-первых, это неудачи, связанные со стилистической перегруженностью и канцеляризацией перевода. В подобных случаях русскоязычный текст становится тяжеловеснее оригинала по форме ввиду использования номинативных конструкций, канцелярских оборотов и отглагольных существительных. Микроконтекст оригинала при этом может быть нейтральным, динамичным и ориентированным на быстрое восприятие, тогда как перевод приобретает более формализованный характер. Подобные решения особенно заметны в передаче заголовков и кратких пояснений, когда текст оригинала опирается на компактную глагольную структуру, а перевод

замещает её статичной и менее гибкой формой. В результате текст документации начинает восприниматься как менее прозрачный и менее естественный для жанра практического технического руководства.

Во-вторых, в корпусе можно выявить стилистические неудачи, обусловленные снижением регистра и разговорной окрашенностью текста перевода. В таких случаях нейтральные или умеренно экспрессивные единицы оригинала передаются через разговорные, публицистически окрашенные либо субъективно усиленные формулы, что не соответствует норме технической документации. Подобные решения не всегда искажают фактическое содержание, однако изменяют тон текста, делая его либо чрезмерно неофициальным, либо стилистически чужеродным по отношению к остальному материалу. Особенно заметны такие сдвиги в тех микроконтекстах, в которых оригинал использует нейтральные средства, а в тексте перевода используются более эмоционально маркированные лексемы. Для реципиента это создаёт ощущение стилистической нестабильности и нарушает единый жанровый режим документации.

Следует подчеркнуть, что стилистические переводческие неудачи не обязательно сопровождаются нарушением предметного содержания микроконтекстов. Во многих случаях фактическая информация передаётся корректно, однако при этом меняется жанрово-регистровый облик текста.

Сопоставление анализируемых документаций показывает, что в переводах MDN Web Docs и Vue.js стилистические решения в большей степени характеризуются устойчивостью и регистровой согласованностью. Даже в тех случаях, когда в переводе встречаются отдельные неудачи, они, как правило, не системны и не меняют общего впечатления от текста как от стилистически цельного документационного материала. В документации React JS, напротив, стилистические неудачи выявляются значительно чаще и затрагивают разные уровни текста: от заголовков и кратких пояснений до развёрнутых обучающих микроконтекстов. Это

позволяет рассматривать их как один из факторов, объясняющих более низкую степень жанровой выверенности перевода именно в данном ресурсе.

Особый интерес представляет тот факт, что значительная часть стилистических неудач в документации React JS возникает во фрагментах, в которых оригинал сочетает техническое содержание с умеренно неформальной, вовлекающей манерой подачи. Перевод подобных микроконтекстов требует тонкой регистровой настройки: чрезмерная формализация приводит к канцеляризации, тогда как попытка приблизить текст к разговорной подаче может вызвать стилистическое снижение или разнородность. Анализ языкового материала показывает, что именно в этих зонах перевод чаще всего оказывается недостаточно устойчивым: в одних случаях оригинальная динамика утрачивается за счёт тяжеловесных номинативных конструкций, в других – нейтральный профессиональный тон смещается в сторону разговорности.

Тем самым стилистические переводческие неудачи оказываются тесно связаны с задачей жанровой адаптации текста документации. Переводчику необходимо сохранить смысл и терминологическую точность, а также воспроизвести тот регистр, который делает документацию удобной для последовательного восприятия (Вовченко, 2025). При отсутствии стилистической выверенности даже корректный по содержанию текст может восприниматься как громоздкий и не вполне профессиональный. Это особенно важно в обучающих разделах, в которых стилистическая организация текста влияет на скорость и комфорт усвоения материала.

3.4.2. Стилистическая перегруженность и канцеляризация

Первым подтипом стилистических переводческих неудач в анализируемом корпусе является стилистическая перегруженность, тесно связанная с канцеляризацией изложения. Подобные неудачи возникают в тех случаях, когда перевод утрачивает присущую документации структурную ясность и динамичность и приобретает более тяжеловесный, книжный характер. Особенно заметно это в заголовках, где англоязычный оригинал, как правило, стремится к краткости и быстрому считыванию, тогда как в переводе появляются громоздкие номинативные конструкции, ослабляющие направляющую роль заголовка.

В документации по веб-разработке заголовок выступает в роли элемента навигации: он должен быстро ориентировать реципиента в содержании раздела и задавать ожидаемый формат дальнейшего материала. В связи с этим стилистическая перегруженность заголовков представляет собой фактор, влияющий на общий режим восприятия текста. Можно предположить, что в подобных случаях переводческое мышление ориентируется преимущественно на внешнюю формальную «серьёзность» технического текста и стремится к максимально книжному способу выражения, тогда как жанровая специфика документации требует большей прозрачности и динамики.

Показателен в этом отношении пример (18):

(18) Detecting common mistakes early.

Раннее обнаружение распространённых ошибок.

В англоязычном оригинале заголовок оформлен в виде герундиальной конструкции с *detecting*, что для документации является типичным способом обозначения динамического процесса. Подобные заголовки задают операционную направленность раздела: реципиент реагирует на такой фрагмент как на указание на то, что именно будет рассматриваться в действии. В переводе динамика

утрачивается, поскольку герундий заменяется отвлечённой номинативной конструкцией «Раннее обнаружение распространённых ошибок». В результате заголовок перестаёт обозначать процессуальность и начинает восприниматься как статичное наименование явления. Подобное смещение – от действия к отвлечённой формуле – и создаёт эффект стилистической перегруженности: русскоязычный вариант звучит более официально, менее гибко и в меньшей степени соответствует жанру практико-ориентированной документации.

(19) Avoiding common rendering problems

Предотвращение возникновения типичных проблем в процессе рендеринга

Аналогичная тенденция наблюдается в (19). Здесь стилистическая перегруженность выражена ещё более отчётливо. Русскоязычный вариант оформлен как номинативная конструкция и дополнительно усложнён за счёт сочетания нескольких абстрактных существительных. В результате заголовок текста становится громоздким и теряет ту композиционную лёгкость, которая присуща оригиналу. Для реципиента это означает, что даже на уровне заголовка восприятие материала требует большего усилия, чем в англоязычной версии. В подобных случаях переводческая личность, по-видимому, тяготеет к стратегии повышения «официальности» текста, однако такой способ перевода вступает в противоречие с жанровой природой документации.

(20) **Prop drilling**

Usually, when we need to pass data from the parent to a child component, we use props. However, imagine the case where we have a large component tree, and a deeply nested component needs something from a distant ancestor component.

Пробрасывание входных параметров

Обычно для передачи данных от родительского компонента в дочерний используются входные параметры. Представьте структуру, в которой будет несколько глубоко вложенных компонентов и потребуется что-то от родительского компонента в глубоко вложенном дочернем.

Сходный случай представляет заголовок раздела в (20). Здесь неудача проявляется в общей стилистической неуклюжести выбранного языкового средства перевода. Англоязычный заголовок предельно краток и строится как компактное обозначение ключевого действия. В переводе вместо сжатой и структурно ясной формы появляется громоздкое отглагольное существительное, которое к тому же создаёт впечатление внутренней терминологической и стилистической неопределённости. Для заголовка документации подобное решение оказывается неудачным: оно не облегчает ориентацию в материале, а, напротив, затемняет её.

Общей чертой микроконтекстов (18)–(20) является то, что в переводе заголовок перестаёт восприниматься как рабочее обозначение действия или практической задачи и начинает звучать как отвлечённая формула. Это особенно важно для документации по веб-разработке, поскольку заголовки в ней выступают в роли смысловых опор. Когда вместо компактного и динамичного обозначения появляется тяжёлая номинативная конструкция, навигационная эффективность заголовка снижается. Реципиент по-прежнему может понять, о чём пойдёт речь, однако стиль для данного жанра изложения становится менее естественным.

Следовательно, стилистическая неудача в подобных микроконтекстах обусловлена неудачным выбором языковых средств перевода и общей стратегией избыточного утяжеления текста. В примерах (18)–(20) перевод демонстрирует склонность к канцеляризации и книжности, тогда как оригинал опирается на краткость, динамику и жанровую функциональность.

Стилистическая перегруженность проявляется в основном тексте документации, прежде всего в тех микроконтекстах, где оригинал ориентирован на простое и последовательное объяснение дальнейших действий.

(21) The very first thing we need to do is to tell the HTML document that we have some CSS rules we want it to use.

Самое первое, что нам нужно сделать, – это сообщить HTML-документу, что у нас есть некоторые правила CSS, которые мы хотим использовать.

В примере (21) предметное содержание в целом сохраняется, однако перевод оказывается стилистически перегруженным. Англоязычный оригинал представляет собой типичное для учебной документации объяснение следующего шага: он построен последовательно, но при этом остаётся достаточно лёгким для восприятия. В переводе же высказывание оформляется через цепочку громоздких конструкций, каждая из которых по отдельности грамматически корректна, но в совокупности создаёт эффект тяжеловесности.

Прежде всего, перегруженность формируется за счёт рамочной формулы «самое первое, что нам нужно сделать». Хотя она воспроизводит общий смысл оригинала, в русскоязычном тексте подобная конструкция звучит избыточно развёрнуто и замедляет движение мысли уже в начале микроконтекста. Далее стилистическую тяжесть усиливает сочетание «сообщить HTML-документу», которое выглядит формально корректным, но не вполне естественным для учебной технической коммуникации. Наконец, завершающая часть «что у нас есть некоторые правила CSS, которые мы хотим использовать» дополнительно осложняет восприятие за счёт длинной придаточной структуры и нагромождения служебных элементов. В результате в (21) утрачивается та степень лаконичности и прозрачности, которая необходима для документации обучающего типа. Фраза остаётся понятной, однако требует от реципиента большего усилия для восприятия и звучит менее естественно,

чем оригинал. Именно в этом и проявляется стилистическая неудача: перевод не нарушает фактического содержания, но передаёт его в более тяжеловесной и в менее жанрово уместной форме.

Наряду с примерами, в которых стилистическая перегруженность проявляется за счёт номинативных конструкций и канцеляризованной подачи, в корпусе можно обнаружить случаи, когда текст утрачивает жанровую естественность из-за буквального и тяжеловесного синтаксического оформления.

(22) This will style any element with a class of special, which is inside a <p>, which comes just after an <h1>. Phew!

Это будет стиль любого элемента с классом special, который находится внутри <p>, который приходит сразу после <h1>, который находится внутри <body>. Уф!

В (22) стилистическая неудача связана прежде всего с тем, что русскоязычный вариант оказывается чрезмерно громоздким и синтаксически неуклюжим. Англоязычный оригинал построен как усложнённая цепочка уточнений, однако внутри жанра учебной документации такая структура остаётся управляемой: она последовательно разворачивает описание и завершается краткой репликой *Phew!*, которая снимает напряжение и маркирует искусственную сложность предыдущей конструкции.

В переводе на русский язык данная цепочка приобретает более тяжёлый характер. Повторяющиеся конструкции «который находится… который приходит… который находится…» делают высказывание монотонным и трудным для восприятия. В результате текст звучит не как живое пояснение в рамках учебного материала, а как стилистически перегруженная и плохо организованная конструкция. Даже при формальной грамматической правильности подобная подача затрудняет чтение и ослабляет естественный темп изложения. Для документации по веб-разработке это особенно существенно, поскольку подобные фрагменты должны не

только сообщать информацию, но и сохранять удобочитаемость при объяснении сложных зависимостей. В (22) такая удобочитаемость оказывается ослабленной: русский вариант требует большего усилия со стороны реципиента и воспринимается как менее гибкий и менее жанрово уместный, чем оригинал.

Показательно и то, что финальное «Уф!», сохраняющееся в переводе, уже не производит того стилистического эффекта, который наблюдается в оригинале. В англоязычном тексте оно работает как комментарий к намеренно усложнённому, но всё ещё управляемому объяснению. В переводе, где основная часть микроконтекста звучит более тяжеловесно, данный элемент оказывается стилистически менее органичным.

3.4.3. Стилистическая сниженность и разговорность

Наряду со стилистической перегруженностью и канцеляризацией в переводе документации по веб-разработке выявляются случаи противоположного характера – стилистическая сниженность и разговорность. Если в первом подтипе перевод оказывался тяжеловесным и чрезмерно книжным, то здесь русскоязычный текст, напротив, смещается в сторону неофициального, разговорного или публицистически окрашенного регистра. Подобные решения не всегда нарушают предметное содержание микроконтекстов, однако они изменяют жанровую тональность документации и снижают степень её стилевой выверенности.

Для англоязычной документации по веб-разработке в целом характерна умеренно неформальная подача. Особенно это заметно в обучающих разделах, где авторы стремятся сочетать техническую точность с доступностью изложения,

избегая как излишней сухости, так и чрезмерной разговорности. Перевод такого текста требует тонкой регистровой настройки: необходимо сохранять живость и естественность текста, не выходя за пределы профессионально-нейтрального стиля. В этой зоне и возникают стилистические неудачи рассматриваемого типа.

Стилистическая сниженность проявляется в тех случаях, когда в переводе используются языковые средства, вносящие в текст нежелательную разговорную, бытовую или публицистическую окраску. Это могут быть разного рода разговорные слова и обороты, экспрессивные характеристики, стилистически сниженные формулы, а также выражения, которые в контексте документации звучат слишком непринуждённо или эмоционально. В результате этого перевод утрачивает регистровую нейтральность и начинает восприниматься как текст, не вполне соответствующий жанру технического руководства.

Следует подчеркнуть, что подобные случаи не обязательно связаны с грубой языковой неуместностью. Гораздо чаще речь идёт о регистровом смещении. Перевод остаётся понятным, иногда даже выглядит «живым» или «естественным» на поверхностном уровне, однако при более внимательном анализе оказывается, что выбранное языковое средство не соответствует общей стилевой норме документации. В таких микроконтекстах возникает ощущение стилистической неустойчивости. Текст звучит либо слишком раскованно, либо просто выбивается из общей тональности соседних фрагментов.

Особую значимость данный подтип стилистических неудач приобретает в тех разделах документации, в которых оригинал сочетает в себе техническое объяснение с умеренно дружелюбной подачей информации. Стремясь сохранить эту доступность, переводчик может прибегать к средствам, которые в русскоязычном техническом дискурсе воспринимаются как сниженные или не вполне уместные. В подобных случаях переводческое стратегирование оказывается направлено на «оживление» текста, однако результатом становится ослабление профессионального регистра.

Материал исследования показывает, что стилистическая сниженность и разговорность чаще всего проявляются на лексическом уровне. Наиболее показательными являются микроконтексты, в которых нейтральные единицы оригинала передаются через разговорно окрашенные слова, публицистически усиленные характеристики или оценочно маркированные формулы. При этом само содержание высказывания может оставаться достаточно близким к оригиналу, однако общий тон перевода меняется: документация начинает звучать менее профессионально и в отдельных случаях – менее убедительно.

Сопоставление анализируемых ресурсов показывает, что данный подтип стилистических неудач наиболее ярко проявляется в документации React JS, где перевод в ряде фрагментов демонстрирует колебание между нейтрально-технической и разговорной подачей. В документациях MDN Web Docs и Vue.js подобные случаи крайне редки и, как правило, не образуют устойчивых серий.

Стилистическая сниженность особенно заметна в тех микроконтекстах, в которых англоязычный оригинал сохраняет нейтральный объяснительный тон, а перевод включает элементы разговорной окрашенности.

(23) However, it misses a crucial requirement: the fact that the Clock sets up a timer and updates the UI every second should be an implementation detail of the Clock.

Проблема в том, что компонент Clock не обновляет себя каждую секунду автоматически. Хотелось бы спрятать логику, управляющую таймером, внутри самого компонента Clock.

В микроконтексте (23) англоязычный оригинал оформлен как нейтральное пояснение. Начало (*it misses a crucial requirement*) задаёт аналитический тон и вводит дальнейшее уточнение, связанное с архитектурной логикой компонента. Вторая часть высказывания также сохраняет профессионально-нейтральный характер:

фрагмент текста *should be an implementation detail* несет соответствующую логику в структуре компонента без разговорной окраски и без субъективизации изложения.

В переводе на русский язык стилистический режим рассматриваемого высказывания меняется. Уже в самом начале «Проблема в том, что…» звучит более прямолинейно и менее аналитично, чем более мягкое *However, it misses a crucial requirement*. Однако наиболее заметный сдвиг возникает во второй части микроконтекста, где нейтральное пояснение заменяется конструкцией «Хотелось бы спрятать…». Подобная форма вносит в текст оттенок разговорности: высказывание начинает звучать как неформальный комментарий или пожелание, а не как объяснение принципа организации компонента.

Для документации по веб-разработке подобная организация высказываний стилистически неудачна, поскольку объяснение архитектурного решения приобретает более бытовой и менее профессиональный характер. Оригинал просто сообщает, как должна быть организована логика компонента, в то время как перевод передаёт неформальное отношение автора к желаемому результату. Вследствие этого ослабляется регистровая нейтральность, а микроконтекст начинает звучать менее строго и менее выверенно, чем оригинал. Стилистическая неудача в (6) связана, таким образом, со снижением регистра: нейтральный документационный тон оригинала в переводе смещается в сторону разговорной окрашенности. При сохранении общего содержания меняется подача материала, что нарушает стилевую цельность текста документации.

Сходные проявления стилистической сниженности наблюдаются и в других микроконтекстах, где нейтральный объяснительный тон оригинала в переводе смещается в сторону более свободной и неформальной подачи.

(24) Next, we’ll make the Clock set up its own timer and update itself every second.

Теперь осталось только установить собственный таймер внутри Clock и обновлять компонент каждую секунду.

В (24) англоязычный оригинал сохраняет характерную для обучающей документации нейтральную направленность: авторы обозначают следующий шаг в работе с компонентом, не прибегая ни к разговорной экспрессии, ни к субъективно окрашенным формулировкам. Конструкция *we'll make the Clock set up its own timer* задаёт последовательное объяснение дальнейшего действия и органично вписывается в общий стиль пошагового изложения. В переводе на русский язык появляется формула «теперь осталось только…», которая вносит в микроконтекст разговорный оттенок. Подобное выражение в учебной коммуникации может звучать естественно в устной речи, однако в тексте документации оно смещает регистр в сторону неформального комментария. Кроме того, лексема «только» придаёт фразе дополнительную субъективную окрашенность, создавая эффект снижения значимости описываемого действия. В результате объяснение следующего этапа начинает восприниматься как более свободная и стилистически сниженная реплика.

Ещё более отчётливо подобное смещение проявляется в микроконтексте (25), где перевод вносит в текст элемент разговорной окрашенности и оценочности.

(25) We want an abstraction that allows us to define this logic in a single place and share it across many components.

Было бы здорово абстрагировать эту функциональность и использовать её в других компонентах.

В англоязычном оригинале высказывание оформлено как концептуально нейтральное пояснение. Первая часть *We want an abstraction* в данном контексте служит способом обозначения проектного принципа: речь идёт о поиске такого решения, которое позволило бы централизованно описать логику и затем использовать её в нескольких компонентах. Тон высказывания остаётся профессионально сдержанным.

В переводе на русский язык данная установка передаётся посредством выражения «Было бы здорово…», которое явно снижает стилевой регистр. Подобная формула характерна скорее для неформального обсуждения или устного объяснения, чем для текста документации. В результате техническое рассуждение приобретает оттенок субъективной оценки, а сама идея абстракции подаётся не как нейтрально сформулированный принцип проектирования, а как неформально выраженное пожелание.

В обоих микроконтекстах – и в (24), и в (25) – предметное содержание в целом сохраняется, однако стилевой регистр высказывания в переводе смещается. Текст становится менее нейтральным и менее жанрово устойчивым. Если в (24) это проявляется посредством разговорной формулы, задающей более свободный тон, то в (25) – через оценочно-разговорное выражение, не соответствующее стилю технической документации. Именно в подобных случаях и обнаруживается стилистическая неудача рассматриваемого подтипа. Перевод стремится к «оживлению» текста, но в результате ослабляет его профессиональную и регистровую цельность.

Анализ стилистических переводческих неудач показал, что, несмотря на их сравнительно небольшую частотность, они заметно влияют на восприятие документации как жанрово цельного и профессионально выверенного текста. Наиболее высокая концентрация таких неудач выявлена в документации React JS, в то время как в MDN Web Docs и особенно в Vue.js они носят более локальный характер. В сформированном для целей исследования корпусе были выделены два основных подтипа стилистических неудач: стилистическая перегруженность и канцеляризация, а также стилистическая сниженность и разговорность. В первом случае перевод утрачивает присущие документации краткость и динамику, приобретая тяжеловесный и более книжный характер; во втором – смещается в

сторону субъективной или разговорной подачи, ослабляя профессиональный регистр текста.

Стилистические неудачи не искажают предметное содержание микроконтекстов, однако изменяют жанрово-регистровый облик документации. В результате перевод может восприниматься как менее естественный, менее последовательный и менее удобный для усвоения, что снижает его функциональную состоятельность как инструмента трансфера знаний.

3.5. Переводческие неудачи синтаксического характера

В ходе осуществления метода сплошного выборки анализа в используемом для целей исследования корпусе было выявлено 68 синтаксических переводческих неудач, что составляет 12,90% от общего объёма материала (527 случаев). Тем самым синтаксические неудачи образуют количественно значимую группу в корпусе и требуют отдельного рассмотрения, поскольку затрагивают способы синтаксической организации документационного текста и влияют на его воспринимаемую прозрачность.

Распределение переводческих неудач синтаксического характера по анализируемым документациям также характеризуется выраженной неравномерностью. Наибольшее их количество зафиксировано в документации React JS – 42 случаев, что составляет 61,76% всех неудач данного типа. Существенно меньшая доля приходится на MDN Web Docs – 20 случаев (29,41%), и минимальное количество выявлено в документации Vue.js – 6 случаев (8,82%). Подобное распределение в целом соотносится с общей структурой корпуса и подтверждает, что

неудачи синтаксического характера наиболее системно проявляются именно в русскоязычной версии документации React JS.

Вместе с тем переводческие неудачи синтаксического характера отличаются от лексико-семантических, прагматических и стилистических неудач тем, что в первых не выявляются устойчивые, повторяющиеся модели. Если в рассмотренных выше типах переводческих неудач можно было говорить о сравнительно легко выделяемых подгруппах с повторяемостью механизмов, то неудачи синтаксического характера в анализируемом материале носят в значительной степени индивидуальный характер. Каждый случай требует отдельного рассмотрения, поскольку связан с конкретной организацией высказывания, структурой предложения, распределением смысловых акцентов и особенностями соотнесенности частей синтаксической конструкции.

Анализ исследовательского корпуса не позволил выявить и чётких закономерностей, связанных с локализацией переводческих неудач синтаксического характера в определённых структурных зонах документации. Подобные случаи встречаются как в обучающих и пояснительных разделах, так и в дефиниционных, инструктивных и описательных микроконтекстах. Иными словами, переводческие неудачи синтаксического характера не привязаны к какому-либо одному типу документационного материала и не образуют серий в пределах конкретных разделов. Это ещё раз подтверждает их неоднородность и вариативность.

Тем не менее, несмотря на отсутствие чётко повторяющихся закономерностей, синтаксические переводческие неудачи объединяет один общий признак. В подобных микроконтекстах перевод сохраняет предметное содержание оригинала, но утрачивает синтаксическую прозрачность, присущую исходному тексту. Это может проявляться в усложнении структуры предложения, неудачном распределении синтаксических связей, чрезмерной тяжеловесности или, напротив, в таком упрощении, которое нарушает логические отношения между частями высказывания. В результате этого текст остаётся формально понятным, однако требует от реципиента большего усилия для интерпретации.

Следует подчеркнуть, что переводческие неудачи синтаксического характера не обязательно сопровождаются лексико-семантическими или терминологическими расхождениями. Во многих случаях неудачным оказывается именно способ синтаксического оформления мысли в переводе. По этой причине данный тип неудач не сводится к стилистической тяжеловесности или прагматическому смещению, хотя на уровне восприятия может пересекаться с ними. Его специфика заключается в том, что синтаксическая организация высказывания перестаёт в полной мере обеспечивать ту степень линейности и структурной ясности, которая необходима для документации обучающего типа. Сопоставление анализируемых документаций показывает, что в переводах MDN Web Docs и Vue.js синтаксические неудачи встречаются значительно реже и, как правило, не оказывают существенного влияния на общее впечатление от текста. В документации React JS, напротив, их количество выше, а сами случаи более разнообразны, что позволяет говорить о меньшей устойчивости синтаксических решений в переводе. Однако и здесь речь идёт не о повторении одного и того же механизма, а скорее о накоплении отдельных, разнотипных отклонений в синтаксической организации микроконтекстов.

Таким образом, переводческие неудачи синтаксического характера представляют собой особую группу в анализируемом корпусе. Они не образуют чётко разграниченных подтипов, не демонстрируют устойчивой привязки к определённым разделам документации и в большинстве случаев требуют индивидуализированного подхода к рассмотрению. Именно поэтому дальнейшее рассмотрение синтаксических неудач будет строиться на основе анализа микроконтекстов, в которых нарушение синтаксической организации перевода влияет на удобочитаемость и интерпретацию документационного текста.

Среди переводческих неудач синтаксического характера, выявленных в документации React JS, особого внимания заслуживают случаи, в которых перевод нарушает внутреннюю связность высказывания и делает синтаксическую

организацию микроконтекста менее прозрачной. Показателен в этом отношении микроконтекст (26).

(26) Catching an event bubbling up from a portal in a parent component allows the development of more flexible abstractions that are not inherently reliant on portals. For example, if you render a <Modal /> component, the parent can capture its events regardless of whether it's implemented using portals.

Перехват событий, всплывающих от портала к родительскому компоненту, позволяет создавать абстракции, которые не спроектированы специально под порталы. Например, вы отрендерили компонент <Modal />. Тогда его события могут быть перехвачены родительским компонентом, вне зависимости от того, был ли <Modal /> реализован с использованием порталов или без них.

Синтаксическая неудача в (26) проявляется прежде всего в том, что перевод ослабляет логико-синтаксические связи, организующие оригинальный микроконтекст. Англоязычный фрагмент выстроен как единое рассуждение: сначала формулируется общее положение, затем через условно-иллюстративную конструкцию *For example, if...* вводится конкретный пример. Связь между условием и следствием выражена прямо и не требует дополнительного восстановления со стороны реципиента.

В переводе условно-следственные отношения выстраиваются другом образом, структура расчленяется. Наблюдается последовательность самостоятельных предложений: «Например, вы отрендерили компонент <Modal />. Тогда его события могут быть перехвачены...». В результате причинно-условные отношения, явно заданные в оригинале, становятся менее явно выраженными. Реципиент уже не считывает взаимоотнесенность условия и следствия в рамках единой синтаксической модели, а вынужден восстанавливать связь между двумя отдельными высказываниями. Дополнительную синтаксическую тяжеловесность создаёт и первая

фраза. Конструкция «Перехват событий, всплывающих от портала к родительскому компоненту...» при всей формальной очевидности правильности воспринимается как менее прозрачная по сравнению с оригиналом. Распределение смысловых связей внутри неё оказывается менее эксплицитно выраженным, а синтаксическая перспектива высказывания становится менее чёткой. Это особенно важно для документации обучающего типа, где сложные концептуальные связи должны быть оформлены максимально управляемым и легко считываемым способом.

Предметное содержание в (26), в целом, сохраняется, однако синтаксическая организация высказывания в переводе становится менее прозрачной, а потому по смыслу более затрудненной. Оригинал задаёт компактное и логически цельное объяснение, тогда как перевод оформляет ту же мысль в более разобщенной и тяжеловесной форме. Именно это и позволяет рассматривать данный микроконтекст как один из наиболее ярких примеров проявления синтаксической переводческой неудачи в документации React JS.

Другой показательный случай связан не столько с расчлененностью синтаксического построения, сколько с перестройкой синтаксических связей внутри сложного высказывания, в результате чего структура в переводе становится громоздкой, а потому и менее прозрачной для понимания. Это наблюдается в микроконтексте (27).

(27) While this is a very important accessibility feature, it is also a technique that should be used judiciously. Use it to repair the keyboard focus flow when it is disturbed, not to try and anticipate how users want to use applications.

Правильное управление фокусом — это важный момент в обеспечении доступности контента, однако стоит применять его с умом. Используйте управление фокусом для восстановления нарушенной последовательности действий при работе с клавиатурой. Не стоит предугадывать желания пользователя и подталкивать его к каким-либо действиям в приложении.

В (27) неудача синтаксического характера проявляется в том, что перевод связан с ослаблением цельности исходного микроконтекста. Текст оригинала характеризуется компактностью высказывания с чётко выраженным противопоставлением и наличием инструктирующей логики: сначала вводится оговорка через *While...,* затем следует основное суждение, после чего формулируется рекомендация в виде четкого противопоставления *Use it... not to....* Таким образом, весь фрагмент текста воспринимается как единая синтаксическая и логическая конструкция. В переводе эта связность не наблюдается. Вторая часть высказывания, рекомендация и ограничение которой давались в оригинале в пределах одной конструкции, представлена двумя самостоятельными структурами: «Используйте управление фокусом...» и «Не стоит предугадывать желания пользователя...». Формально смысл при этом передается, однако синтаксическая организация становится менее прозрачной. Противопоставление, выраженное в оригинале в пределах одной фразы, в переводе перестаёт восприниматься как единое целое и требует со стороны реципиента принятия дополнительных усилий по его установлению.

Кроме того, в переводе используется более тяжеловесная и «бюрократизированная» синтаксическая форма в первой части микроконтекста: «Правильное управление фокусом – это важный момент в обеспечении доступности контента...». Данная конструкция передает в переводе содержание оригинала, но звучит более громоздко по сравнению с *this is a very important accessibility feature*, что придает менее ясный ритм дальнейшему объяснению. В сочетании с последующей расчлененностью инструкции это усиливает общий эффект синтаксической перегруженности.

Особенно заметно это в части «для восстановления нарушенной последовательности действий при работе с клавиатурой». Русскоязычный вариант организован посредством постепенного присоединения компонентов структуры и

требует, соответственно, более медленного синтаксического развёртывания для понимании, тогда как *to repair the keyboard focus flow when it is disturbed* в тексте оригинала воспринимается как более компактная линейная структура. В результате перевод оказывается менее удобным для быстрого восприятия, хотя общая логика высказывания остаётся доступной.

Сходный тип синтаксической неудачи наблюдается и в тех случаях, когда перевод как бы перераспределяет синтаксические связи, прослеживаемые в оригинале, в результате чего объяснение становится менее последовательным. Это можно проследить в микроконтексте (28).

(28) React.lazy takes a function that must call a dynamic import (). This must return a Promise which resolves to a module with a default export containing a React component.

React.lazy принимает функцию, которая должна вызвать динамический import (). Результатом возвращённого Promise является модуль, который экспортирует по умолчанию React-компонент (export default).

В (28) англоязычный оригинал организован как последовательное объяснение из двух связанных предложений. Сначала задаётся исходное условие: *React.lazy* принимает функцию, которая должна вызвать динамическую функцию *import ()*. Затем во втором предложении эта логика продолжает развиваться: местоимение *This* отсылает к результату вызова *dynamic import ()* и вводит следующее звено объяснения – необходимость возврата *Promise*. Синтаксическая структура оригинала линейна и выстроена по принципу пошагового наращивания информации.

В переводе эта строгая последовательность уже не прослеживается. Вторая часть микроконтекста конструктивно меняется: «Результатом возвращённого Promise является модуль...». Такое переводческое решение позволяет поменять синтаксическую перспективу высказывания: вместо естественного продолжения предыдущей мысли появляется более тяжеловесно сформулированная структура.

При этом исчезает степень прямой связности, которая была заложена в оригинале благодаря формулировке *This must return a Promise*.... Реципиент текста перевода уже не так легко может проследить, что именно к чему относится и в каком порядке разворачивается логика объяснения. Помимо этого, наблюдается тяжеловесность структуры русского предложения. Часть «Результатом возвращённого Promise является модуль, который экспортирует по умолчанию...» строится путем соединения нескольких последовательно вложенных компонентов и требует от потенциального реципиента при восприятии более сложных умственных манипуляций по сравнению с получателем англоязычного текста оригинала.

Таким образом, в тексте перевода микроконтекста (28) предметное содержание в целом сохраняется, однако перевод нарушает исходную линейность объяснения и делает синтаксическую структуру менее прозрачной для восприятия. Текст оригинала позволяет последовательно разворачивать причинно-логическую цепочку, тогда как перевод оформляет её в более статичной и усложнённой форме.

Приведем далее пример того, как в переводе по сравнению с текстом оригинала изменяется логика высказывания, вследствие чего объяснение становится нелинейным и менее удобным для восприятия.

(29) When you compare objects with ===, JavaScript checks whether they reference the same object, not whether they contain the same values.

Не то, что они содержат одинаковые значения, а то, ссылаются ли они на один и тот же объект, JavaScript проверяет при сравнении объектов с помощью ===.

В (29) англоязычный оригинал синтаксически организован предельно прозрачно. Сначала задаётся условие – *When you compare objects with* ===, затем формулируется основное утверждение о том, что именно проверяет JavaScript, после чего следует уточняющее противопоставление *not whether they contain the same*

values. Логика развёртывания высказывания последовательна: от действия к его результату, затем к уточнению.

В переводе переводчик радикально перестроил исходную структуру. На первый план вынесено противопоставление «Не то, что они содержат одинаковые значения, а то, ссылаются ли они на один и тот же объект…», тогда как подлежащее и предикат – «JavaScript проверяет» – появляются лишь в конце фразы. В результате реципиент перевода сталкивается с выражением, грамматическая основа которого становится очевидной только после достаточно протяжённого фрагмента. Такая организация высказывания затрудняет его последовательное восприятие и требует большего усилия для установления синтаксических связей между элементами предложения.

Особую тяжеловесность создаёт то, что в переводе предложение начинается с части, которая в оригинале занимает зависимую, уточняющую позицию. В англоязычном микроконтексте сначала сообщается, какое действие выполняет JavaScript, и лишь затем идёт уточнение. В переводе этот порядок нарушается: реципиенту сначала предлагается противопоставительная конструкция, а уже потом предлагается основное предикативное ядро. Для объяснения базового принципа работы оператора сравнения такое изменение структуры предложения представляется неудачной, поскольку снижает прозрачность изложения.

3.6. Гибридные переводческие неудачи и их кумулятивный эффект

Отдельного рассмотрения заслуживают микроконтексты, в которых переводческая неудача проявляется одновременно в нескольких плоскостях. В таких случаях реципиент сталкивается с совокупностью расхождений, взаимно

усиливающих друг друга. В подобных микроконтекстах различные типы неудач действуют в комплексе, формируя кумулятивный эффект.

В рамках настоящего исследования к гибридным случаям было отнесено 76 микроконтекстов (14,42% от общего количества), из которых 34 выявлены в документации React JS (44,74%), 26 – в MDN Web Docs (34,21%), и 16 – в Vue.js (21,05%). Данная статистика показывает, что гибридные неудачи встречаются во всех анализируемых ресурсах, хотя их характер и степень выраженности различаются. В документации React JS они чаще связаны с лексико-семантическим и прагматическим аспектами, в MDN Web Docs – с сочетанием прагматических факторов, тогда как в Vue.js подобные случаи, как правило, носят более локальный характер.

Специфика гибридных неудач состоит в том, что каждая из составляющих по отдельности может восприниматься как малозаметное расхождение. Однако в пределах одного микроконтекста такие неудачи носят накопительный характер. Например, терминологическая неустойчивость может сочетаться с прагматическим ослаблением инструкции, а стилистическая тяжеловесность – с синтаксической неудачей, затрудняющей восприятие высказывания. В результате смещение коммуникативного эффекта оказывается более существенным, чем при наличии только одной неудачи: реципиенту сложнее восстановить исходную логику текста, определить статус высказывания и соотнести его с общей структурой документации. Гибридные неудачи. позволяют увидеть, что переводческая неудача в документации по веб-разработке часто носит комплексный характер. В таких микроконтекстах особенно отчётливо проявляется зависимость между различными уровнями организации текста: лексический выбор влияет на прагматический эффект, синтаксическая структура – на стилистическое восприятие, а невербальные элементы могут быть связаны с коммуникативной цельностью фрагмента. Далее будут рассмотрены наиболее яркие примеры подобных сочетаний.

Гибридная переводческая неудача прослеживается в микроконтексте (30), в котором одновременно затрагиваются лексико-семантический, прагматический и синтаксический аспекты.

(30) Code-splitting your app can help you “lazy-load” just the things that are currently needed by the user, which can dramatically improve the performance of your app. While you haven’t reduced the overall amount of code in your app, you’ve avoided loading code that the user may never need, and reduced the amount of code needed during the initial load.

Хоть вы и не уменьшите общий объём кода вашего приложения, но избежите загрузки кода, который может никогда не понадобиться пользователю и уменьшите объём кода, необходимого для начальной загрузки.

В (30) прежде всего обращает на себя внимание то, что перевод охватывает только вторую часть исходного микроконтекста. Первая фраза, в которой вводится ключевое понятие *code-splitting* и объясняется его связь с *lazy-load* (букв. «ленивая загрузка»), в переводе отсутствует. Уже на этом этапе нарушается содержательная и коммуникативная целостность фрагмента: реципиент получает фрагмент рассуждения без исходного пояснения, необходимого для его полноценного понимания. По этой причине последующее высказывание воспринимается менее связно, чем в оригинале, а логика объяснения оказывается ослабленной.

Лексико-семантический аспект неудачи связан прежде всего с выпадением концептуально значимых единиц. В оригинале микроконтекст строится как пояснение механизма: сначала называется действие (*code-splitting your app*), затем раскрывается его практический результат (*lazy-load just the things that are currently needed by the user*), после чего формулируются последствия для производительности. В переводе данная связка не воспроизводится и реципиенту предлагается только

часть рассуждения без указания на то, о каком именно механизме идёт речь. В результате смысловой каркас фрагмента заметно ослабляется.

Прагматическая неудача проявляется в том, что перевод утрачивает объяснительную направленность оригинала. Англоязычный микроконтекст выполняет информативную и обучающую задачу: он вводит новый для реципиента принцип и сразу же показывает, зачем он нужен и как работает. В переводе же остаётся лишь часть рассуждения, которая без предшествующего пояснения воспринимается как фрагмент общего комментария, а не как последовательное объяснение. Тем самым нарушается управляемость обучающего сценария: в результате реципиент не получает цельную когнитивную траекторию, заданную оригиналом.

Наконец, синтаксический аспект неудачи связан с тем, что даже сохранившаяся часть высказывания оформлена в переводе более тяжеловесно. Конструкция «Хоть вы и не уменьшите общий объём кода вашего приложения, но избежите загрузки кода...» воспринимается как перегруженная и требует дополнительного усилия для синтаксического развёртывания. Особенно заметно это в заключительной части, где последовательно нанизываются зависимые компоненты: «…кода, который может никогда не понадобиться пользователю и уменьшите объём кода, необходимого для начальной загрузки». В результате высказывание теряет прозрачность, характерную для английского микроконтекста.

Ещё один гибридный случай представлен в микроконтексте (31), где одновременно проявляются прагматическая и синтаксическая неудачи.

(31) A good place to start is with routes. Most people on the web are used to page transitions taking some amount of time to load. You also tend to be re-rendering the entire page at once so your users are unlikely to be interacting with other elements on the page at the same time.

Часто таким удобным местом оказываются маршруты. Большинство интернет-пользователей привыкли к задержкам во время переходов между страницами. Поэтому и вам может быть выгодно повторно отрендерить всю страницу целиком. Это не позволит пользователям взаимодействовать с другими элементами на странице, пока происходит обновление.

В (31) перевод в целом воспроизводит общее направление рассуждения, однако делает это с заметным смещением логики оригинала. Англоязычный микроконтекст строится как последовательное объяснение: сначала указывается, что удобной отправной точкой могут быть маршруты, затем поясняется, почему именно они представляют собой удачное решение, и, наконец, формулируется вероятное следствие – пользователи, скорее всего, не будут взаимодействовать с другими элементами страницы одновременно с её полной перерисовкой. Оригинал, таким образом, не навязывает жёсткого вывода, а предлагает аргументированное обоснование.

В переводе же уже первая фраза подвергается изменению: *A good place to start is with routes* переведена как «Часто таким удобным местом оказываются маршруты». Оригинал указывает на конкретную стратегическую рекомендацию: начать с маршрутов; перевод же оформляет высказывание как более общее наблюдение. Тем самым ослабляется направляющая логика микроконтекста.

Прагматический сдвиг особенно заметен в заключительной части. В оригинале высказывание *your users are unlikely to be interacting with other elements on the page at the same time* сохраняет оттенок вероятности, неточности и описывает типичную ситуацию. В переводе эта мысль звучит как факт: «Это не позволит пользователям взаимодействовать с другими элементами на странице, пока происходит обновление». В результате рассуждение преобразуется в практически категоричное утверждение о невозможности взаимодействия. Это меняет коммуникативную

установку фрагмента: вместо объяснения типичного пользовательского сценария реципиент получает более прямое следствие.

Неудача синтаксического характера проявляется в том, что перевод оформляет исходное рассуждение как последовательность разрозненных утверждений, связь между которыми менее прозрачна, чем в оригинале. Особенно это заметно в переходе от «Поэтому и вам может быть выгодно повторно отрендерить всю страницу целиком» к «Это не позволит пользователям взаимодействовать...». Такая организация текстового развертывания ослабляет внутреннюю причинно-следственную связь. В оригинале выстраивается единое объяснение внутри одной логической цепочки, тогда как в переводе наблюдается её расчлененность на отдельные выводы.

Анализ гибридных переводческих неудач показал, что наиболее сильное воздействие на реципиента производят те микроконтексты, в которых одновременно проявляются несколько типов неудач. В подобных случаях переводческая неудача не локальна: лексико-семантические, прагматические, стилистические и синтаксические сдвиги начинают действовать совместно и оказывают кумулятивный эффект.

Гибридные переводческие неудачи особенно показательны тем, что позволяют увидеть текст документации как многоуровневую коммуникативную систему. В таких микроконтекстах нарушение на одном уровне почти неизбежно отражается на других: синтаксическая тяжеловесность усиливает прагматическое смещение, лексико-семантическое выпадение ослабляет объяснительную функцию, а рассинхронизация с невербальными элементами делает фрагмент менее цельным.

Выводы по главе 3

Сопоставительное исследование англоязычных оригиналов и переводов на русский язык официальной документации React JS, MDN Web Docs и Vue.js показало, что переводческие неудачи в документации по веб-разработке носят системный характер и затрагивают различные аспекты текста. Их возникновение связано не столько с особенностями отдельных переводческих решений, сколько с общим характером переводческого стратегирования, степенью последовательности реализации принятой стратегии, а также с тем, насколько перевод позволяет сохранять текст документации в его исходной функции, т.е. как инструмент структурированного трансфера знаний.

Количественный и качественный анализ языкового материала позволил установить, что наибольшая концентрация переводческих неудач наблюдается в документации React JS. Данное обстоятельство не может быть сведено исключительно к объёму анализируемого материала. Существенную роль здесь играют, видимо, менее устойчивый характер переводческих решений, неоднородность в воспроизведении коммуникативных параметров оригинала, а также асинхронность обновления англоязычной и русскоязычной версий документации. В документациях MDN Web Docs и Vue.js переводческие неудачи также фиксируются, однако в целом они носят более локальный характер и реже образуют повторяющиеся серии, затрагивающие однотипные микроконтексты.

В лексико-семантическом измерении наиболее значимыми оказались неудачи, связанные с передачей терминов. Проведённый анализ показал, что здесь особенно отчётливо проявляется расхождение между формальным сохранением предметного содержания и изменением когнитивной организации текста. Сужение или расширение значения лексем, терминологическая вариативность, конкуренция русскоязычных и англоязычных обозначений и снижение прозрачности отдельных терминов приводят к фрагментации понятийной картины. В результате перевод

сохраняет общую информативность, однако в меньшей степени обеспечивает формирование устойчивых смысловых связей и снижает эффективность дальнейшего освоения материала реципиентом.

Установлено, что в прагматическом измерении переводческие неудачи наиболее последовательно проявляются в передаче модальной семантики, директивности, эмоционального тона и невербальных элементов документации. При сохранении фактического содержания микроконтекста изменяется характер взаимодействия реципиента с текстом. Ослабление или усиление модальности, смещение инструктивного статуса высказывания, нейтрализация поддерживающего эмоционального компонента, а также рассогласование между вербальными и невербальными элементами документации значительно изменяют коммуникативный эффект. Прагматические переводческие неудачи обнаруживают себя прежде всего как функциональные расхождения, влияющие на способ интерпретации текста и на характер его усвоения.

Анализ стилистических переводческих неудач показал, что существенную роль в переводе документации играет жанрово-регистровая организация текста. В рамках исследования были выделены два основных направления стилистических неудач: стилистическая перегруженность и канцеляризация, с одной стороны, и стилистическая сниженность и разговорность – с другой. В первом случае перевод утрачивает динамику, краткость и композиционную прозрачность, приобретая более тяжеловесный вид; во втором – смещается в сторону субъективной, разговорной или публицистически окрашенной подачи, не вполне соответствующей норме технической документации. При этом предметное содержание высказывания в большинстве случаев сохраняется, однако меняется жанрово-регистровый облик текста, что ослабляет его профессиональную цельность и снижает удобство восприятия.

В синтаксическом измерении переводческие неудачи обнаруживают иную специфику. В отличие от лексико-семантических, прагматических и стилистических

неудач, они не образуют чётко разграничиваемых подтипов и не демонстрируют устойчивой привязки к определённым структурным зонам документации. Каждый случай в значительной степени индивидуален и связан с особенностями синтаксической организации конкретного высказывания. Вместе с тем проведённый анализ показал, что неудачи синтаксического характера объединяет общий признак: перевод сохраняет предметное содержание оригинала, однако утрачивает синтаксическую прозрачность, необходимую для быстрого и адекватного восприятия текста документации. Это проявляется в расчлененности структуры, усложнении синтаксической перспективы и ослаблении логико-синтаксических связей между частями высказывания.

Помимо этого, был выявлен ряд гибридных переводческих неудач. Их значимость определяется тем, что в пределах одного микроконтекста неудачи проявляются сразу в нескольких аспектах: лексико-семантическом, прагматическом, стилистическом и синтаксическом. В таких случаях действие каждой отдельной неудачи усиливается за счёт наложения на другие. Выпадение концептуально значимых единиц ослабляет объяснительную функцию, синтаксическая тяжеловесность затрудняет восприятие, прагматический сдвиг изменяет статус высказывания, а стилистическая невыверенность нарушает жанровую цельность текста. Гибридные случаи наиболее наглядно демонстрируют, что переводческая неудача в тексте перевода документации по веб-разработке представляет собой, как правило, комплексное явление.

Проведённое исследование позволяет заключить, что перевод документации по веб-разработке требует сохранения предметной информации, а также той сложной системы связей, которая обеспечивает её поэтапное, управляемое и жанрово адекватное усвоение. Переводческая неудача в данном типе текста проявляется прежде всего как функциональное расхождение между оригиналом и переводом, способное нарушать когнитивную, коммуникативную и жанровую организацию документации. Следовательно, качество перевода в данной сфере должно

оцениваться не только по критерию содержательной эквивалентности, но и по степени сохранения документации как целостного механизма осуществления трансфера знаний.

Заключение

Подводя итоги проведённого исследования, следует отметить, что перевод англоязычной документации по веб-разработке на русский язык представляет собой особую область переводческой деятельности, в которой пересекаются задачи точной передачи специального знания, сохранения функциональной организации текста и обеспечения его доступности для профессионального реципиента.

Актуальность такой постановки вопроса определяется тем, что официальная документация по веб-разработке выступает одним из основных каналов знакомства русскоязычных специалистов с современными веб-технологиями, а качество её перевода существенным образом влияет на характер усвоения терминологии, понимание технологических концепций и корректность дальнейшего применения описываемых решений.

Было установлено, что изучение перевода документации по веб-разработке требует обращения к более широкой исследовательской рамке, включающей трансфер знаний, профессиональную коммуникацию, специфику технического текста и функционально ориентированное понимание качества перевода.

Проведённый анализ показал, что трансфер знаний в сфере информационных технологий представляет собой сложный процесс, включающий сохранение, структурирование, трансформацию, адаптацию и практическое освоение информации. Веб-разработка занимает в этом процессе особое место, поскольку характеризуется высокой скоростью изменений, постоянным обновлением инструментов и необходимостью быстрой интеграции специалистов в новые профессиональные контексты. В такой ситуации документация становится одним из центральных механизмов передачи знаний, а её перевод условием включения этих знаний в русскоязычную профессиональную среду.

Был уточнён статус документации по веб-разработке как особой разновидности специального текста. Установлено, что она сочетает признаки технического и учебного текста, выполняет одновременно справочную, инструктивную и обучающую функции, ориентируется на практическое действие реципиента и характеризуется структурированностью, логической организацией материала, активным использованием примеров кода, гиперссылок и пояснительных блоков. Это позволило рассматривать документацию по веб-разработке как интернет-текст, в котором передача знания тесно связана с организацией профессиональной деятельности реципиента в цифровом формате. На этом основании был сделан вывод о том, что качество её перевода определяется способностью текста сохранять прозрачность, инструктивность, логическую последовательность, пригодность для практического применения и эффективного внедрения в цифровые практики.

В рамках исследования были разграничены понятия переводческой ошибки и переводческой неудачи. Обзор переводоведческой литературы показал, что понятие переводческой ошибки традиционно используется для описания локальных отклонений перевода от оригинала, выявляемых в разных языковых аспектах. Вместе с тем, применительно к документации по веб-разработке данное понятие обладает недостаточной объяснительной силой для описания выявляемых сдвигов в переводе, поскольку значительная часть отклонений не сводится к явным языковым нарушениям, но при этом снижает коммуникативную эффективность текста, ослабляет его обучающую направленность, нарушает жанровые ожидания адресата и препятствует трансферу специальных знаний. Это сделало необходимым обращение к более широкому понятию переводческой неудачи.

Методологическим основанием такого терминологического выбора стали интегративное переводоведение и коммуникативно-функциональный подход, позволившие рассматривать перевод как сложную динамическую систему, зависящую от языковых, когнитивных, прагматических, дискурсивных и деятельностных факторов. Существенным для дальнейшего анализа представляется

обращение к понятию переводческого стратегирования, в рамках которого переводческие неудачи были интерпретированы как следствия сбоев в реализации переводческого стратегирования, то есть как результат непоследовательности или функциональной несогласованности переводческих решений.

Рассмотрение основных характеристик документации к технологиям React JS, Vue.js и MDN Web Docs позволило показать, что все три ресурса представляют собой значимые формы организации специального знания в сфере веб-разработки, но при этом различаются как по предметному охвату, так и по структуре, коммуникативной организации и степени стилистической однородности.

React JS характеризуется как одна из фундаментальных библиотек современной фронтенд-разработки, документация которой построена последовательно, от базовых понятий к более сложным аспектам, но русскоязычная версия при этом содержит устаревшие разделы, неполные переводы и задержки в обновлениях.

Vue.js описывается как прогрессивный фреймворк, сочетающий простоту, гибкость и постепенную внедряемость; его документация характеризуется высокой диалогичностью, обучающей направленностью и системной организацией, хотя русскоязычная версия также демонстрирует несинхронность обновлений и стилистическую неоднородность.

MDN Web Docs представляется как крупнейший открытый справочник по веб-технологиям, сочетающий учебную и справочную функции и выполняющий роль универсального репозитория знаний.

Во всех трёх рассматриваемых источниках документация выступает как средство профессиональной социализации, формирующее единые способы интерпретации технологии, терминологические нормы и практики использования технических знаний.

Проведённый анализ позволяет установить, что переводческие неудачи в данной сфере проявляются в разных аспектах текстовой организации и могут носить системный характер. Наибольшая концентрация таких неудач зафиксирована в

документации React JS, что подтверждает выдвинутую гипотезу о её меньшей жанровой и коммуникативной выверенности по сравнению с переводами MDN Web Docs и Vue.js. Последние также содержат переводческие неудачи, однако в целом они проявляются менее системно и реже образуют повторяющиеся ряды.

В лексико-семантическом аспекте рассмотрения переводческие неудачи чаще всего связаны с передачей терминов и элементов метаязыка документации. Подобные случаи проявляются в смещении семантического акцента, нарушении устойчивости терминологических соответствий, избыточной конкретизации или, напротив, неоправданном расширении значения. Они не всегда приводят к грубому искажению предметного содержания, однако затрудняют корректную интерпретацию профессиональных понятий и нарушают предсказуемость лексической организации документационного текста. В условиях, когда терминологическая точность непосредственно связана с успешностью действий адресата, такие неудачи оказываются особенно значимыми.

Прагматические переводческие неудачи интерпретируются как неудачи, связанные с несоответствием перевода коммуникативному намерению оригинала и ожиданиям адресата. Они связаны со степенью директивности и инструктивности текста, распределением ответственности между автором и пользователем, выбором способов маркирования ограничений и предостережений, а также моделированием сценариев использования технологии веб-разработки. Прагматическая неудача возможна и при отсутствии очевидных лексико-грамматических нарушений, поскольку она проявляется в недооценке того, для чего текст предназначен и каким образом он должен работать в реальной коммуникации. Тексты документации по веб-разработке должны быть ориентированы на предсказуемое действие пользователя и предупреждение неверных интерпретаций.

Стилистические переводческие неудачи, исследуемые в русле стилистики дискурса, показали, что даже при сохранении предметного содержания перевод может изменять жанрово-регистровый облик документации и тем самым снижать её

функциональную состоятельность. Осуществлённый стилистический анализ текстов документации и их переводов позволил выделить два основных подтипа таких неудач: стилистическую перегруженность и канцеляризацию, а также стилистическую сниженность и разговорность. В первом случае перевод утрачивает присущие документации краткость и динамику, во втором форма представления информации смещается в сторону разговорной или субъективно окрашенной подачи, ослабляя профессиональный регистр текста. Интенсивность проявления стилистических неудач возрастает в тех зонах, в которых текст оригинала сочетает техническое содержание с умеренно неформальной, вовлекающей подачей.

Переводческие неудачи синтаксического характера осмысливаются как неудачи, в результате которых снижаются прозрачность инструкций, когнитивная доступность материала и управляемость чтения. В документации по веб-разработке даже формально допустимый синтаксический выбор может затруднять пошаговую интерпретацию, размывать причинно-следственные связи, усложнять восприятие условий применимости действия и тем самым снижать адекватность текста по отношению к коммуникативной ситуации. Синтаксические параметры рассматриваются в связи с этим как способ структурирования знания и управления действиями адресата.

Отдельно были выделены гибридные случаи, в которых переводческая неудача проявляется одновременно в нескольких аспектах. Подобные микроконтексты заслуживают особого рассмотрения, поскольку различные типы неудач действуют в комплексе и формируют кумулятивный эффект. В результате даже сравнительно небольшие по отдельности расхождения в совокупности создают более заметное снижение функциональной результативности перевода, усиливают когнитивную нагрузку на реципиента и затрудняют усвоение представленного знания.

Таким образом, в результате проведённого исследования подтверждена исходная гипотеза. При переводе англоязычной документации по веб-разработке на русский язык действительно выявляются переводческие неудачи, препятствующие

осуществлению трансфера специального знания и снижающие качество перевода. Данные неудачи затрагивают разные аспекты организации документации и во многих случаях связаны с недостаточной последовательностью переводческого стратегирования. Исследование показало, что категория переводческой неудачи обладает большей объяснительной силой, чем категория переводческой ошибки в её традиционном понимании, поскольку позволяет учитывать не только само наличие несоответствия, но и его последствия для функционирования текста в профессиональной среде пользователя.

Перспектива данного исследования определяется возможностью дальнейшего изучения перевода специальной документации в цифровой среде, расширения корпуса анализируемых технологий, а также более детального рассмотрения связи между типом документации, параметрами переводческого стратегирования и характером функциональных сбоев в переводе. Представляется целесообразным продолжить исследование как на материале других видов ИТ-документации, так и в направлении разработки практических рекомендаций по переводу и локализации технических текстов, ориентированных на сохранение их обучающей, справочной и инструктивной результативности.

Список литературы

1. Абсалямова, А. А. Разработка и использование веб-API: обзор технологий и инструментов / А. А. Абсалямова, Ж. Ф. Сафиканова. – Текст : непосредственный // Современная российская наука: актуальные вопросы, достижения и инновации. – 2023. – С. 19–21.

2. Авцынова, К. В. Специфика и особенности перевода технических текстов / К. В. Авцынова. – Текст : непосредственный // Вестник науки. – 2023. – № 6 (63). – С. 796–798.

3. Алексеева, Л. М. Трансфер знания: инновации и технологии / Л. М. Алексеева, С. Л. Мишланова. – Пермь, 2022. – 206 с. – Текст : непосредственный.

4. Алипичев, А. Ю. Темпоральность перевода технического текста / А. Ю. Алипичев, А. Н. Кузнецов, Н. А. Сергеева. – Текст : непосредственный // Вестник Череповецкого государственного университета. – 2024. – С. 79–89.

5. Аманов, М. Э. Адаптация и перспективы развития традиционных образовательных технологий в условиях цифровой трансформации образования / М. Э. Аманов, А. А. Нурмухаммедова. – Текст : непосредственный // Colloquium-journal. – 2022. – С. 23–27.

6. Антонов, С. А. Обзор современных библиотек для разработки интерфейса веб-приложения / С. А. Антонов, А. А. Вуколов, К. А. Кононыхина. – Текст : непосредственный // Вестник науки. – 2024. – № 5. – С. 477–504.

7. Арсеньев, И. В. Этапы разработки веб-приложения / И. В. Арсеньев, М. О. Роднянский, Н. Р. Сутягин. – Текст : непосредственный // Актуальные вопросы современной науки и образования: сборник статей XXII Международной научно-практической конференции. – Пенза : МЦНС «Наука и Просвещение», 2022. – 254 с.

8. Баженова, Е. А. Дискурсивный вектор функциональной стилистики / Е. А. Баженова, М. А. Ширинкина. – Текст : непосредственный // Стылістыка: мова,

маўленне і тэкст : матэрыялы V Міжнар. навук. канф., Мінск, 21–22 красавіка 2022 г. – Мінск : БДУ, 2022. – С. 23–30.

9. Базжина, В. А. Трансфер знаний в рамках клинического подхода к обучению в современном университете (на примере Центра прикладных экономических исследований СПбГУ) / В. А. Базжина, О. С. Муравьева, А. А. Стрелкова. – Текст : непосредственный // Инновационная экономика. – 2023. – № 1 (291). – С. 85–94.

10. Байдыбеков, А. А. Современные фреймворки для разработки WEB-приложений / А. А. Байдыбеков, Р. Г. Гильванов, И. А. Молодкин. – Текст : непосредственный // Интеллектуальные технологии на транспорте. – 2020. – № 4. – С. 23–29.

11. Баринова, И. А. Переводческие ошибки: коммуникативные vs когнитивные / И. А. Баринова, И. Г. Овчинникова. – Текст : непосредственный // Научный вестник Воронежского государственного архитектурно-строительного университета. Лингвистика и межкультурная коммуникация. – 2015. – № 2. – С. 93–100.

12. Бархударов, Л. С. Язык и перевод (Вопросы общей и частной теории перевода) / Л. С. Бархударов. – Москва : Международные отношения, 1975. – 240 с. – Текст : непосредственный.

13. Бедный, А. Б. Основы организации университетского трансфера знаний: учебное пособие / А. Б. Бедный. – Н. Новгород : Изд-во ННГУ, 2010. – 40 с. – Текст : непосредственный.

14. Белашова, В. Г. Информационные технологии и конфиденциальность / В. Г. Белашова. – Текст : непосредственный // Форум молодых ученых. – 2017. – № 12 (16). – Саратов : ООО «Институт управления и социально-экономического развития». – С. 154–161.

15. Божева, Е. О. Обучающие эффекты экспорта в Российской софтверной индустрии / Е. О. Божева. – Текст : непосредственный // Стратегические решения и риск-менеджмент. – 2018. – № 2 (107). – С. 38–49.

16. Бондаренко, С. О. Современные интерактивные веб-приложения – построение пользовательского интерфейса с React / С. О. Бондаренко. – Текст : непосредственный // Вестник науки и образования. – 2018. – № 5 (41). – С. 46–48.

17. Бочавер, С. Ю. Концептуализация трансфера и перевода в современной лингвистике / С. Ю. Бочавер, В. В. Фещенко. – Текст : непосредственный // Слово.ру: Балтийский акцент. – 2017. – Т. 8, № 3. – С. 7–29.

18. Буран, А. Л. К вопросу об основных лингвистических характеристиках технического текста / А. Л. Буран. – Текст : непосредственный // Вестник Томского государственного педагогического университета. – 2012. – № 4 (119). – С. 97–99.

19. Васильева, Л. И. Анализ результативности проведения хакатонов и идеатонов / Л. И. Васильева, Э. И. Дяминова, А. С. Филиппова. – Текст : непосредственный // Образовательные ресурсы и технологии. – 2024. – № 3 (48). – С. 7–17.

20. Васюченок, Л. П. Трансфер технологий как экономический ресурс / Л. П. Васюченок. – Текст : непосредственный // Экономическая наука сегодня. – 2015. – № 3. – С. 15–22.

21. Власова, Н. С. Особенности подготовки веб-разработчиков в высшей школе / Н. С. Власова. – Текст : непосредственный // Проблемы современного педагогического образования. – Ялта : Крымский федеральный университет, 2021. – № 3. – С. 82–85.

22. Вовченко В. Е., Поликарпов А.М. Типология переводческих неудач в аспекте качества перевода на русский язык англоязычных текстов документации по веб-разработке / В. Е. Вовченко, А. М. Поликарпов. – Текст : непосредственный // Вестник ПНИПУ. Проблемы языкознания и педагогики. – 2025. – №3. – С. 8–20.

23. Вовченко В. Е. Стилистические неудачи в переводе англоязычных текстов документации по веб-разработке на русский язык / В. Е. Вовченко. – Текст : непосредственный // Вестник Тверского государственного университета. Серия «Филология». – 2026. – №2. – С. 242–248.

24. Вовченко В. Е. Влияние англоцентризма на перевод документации по веб-разработке на русский язык / В. Е. Вовченко. – Текст : непосредственный // Вестник Северного (Арктического) федерального университета. Серия: Гуманитарные и социальные науки. – 2026. – Т. 26, №2. – С. 101–108.

25. Выволокина, Ю. Е. Переводческие ошибки в аспекте теории и практики / Ю. Е. Выволокина. – Текст : непосредственный // Студенческая научная сессия. – Воронежский государственный университет, 2020. – С. 1–8.

26. Газизуллин, Н. И. Разработка прогрессивного веб-приложения с помощью технологии PWA / Н. И. Газизуллин, И. Е. Плещинская. – Текст : непосредственный // StudNet. – Москва : Общество с ограниченной ответственностью «Электронная наука», 2020. – С. 620–625.

27. Гек, Д. К. Angular как средство создания нативного приложения / Д. К. Гек, Д. В. Тихоненко. – Текст : непосредственный // Актуальные проблемы авиации и космонавтики. – Красноярск : Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева, 2021. – С. 486–487.

28. Гительман, Л. Д. Технология ускоренного трансфера знаний для опережающего обучения специалистов цифровой экономики / Л. Д. Гительман, М. В. Кожевников, О. Б. Рыжук. – Текст : непосредственный // Экономика региона. – 2020. – № 16 (2). – С. 435–448.

29. Голубева, О. В. Переводческая деятельность как способ познания мира / О. В. Голубева. – Текст : непосредственный // Этнопсихолингвистика. – 2022. – № 1 (8). – С. 21–34.

30. Горбунов, А. Ю. Структура и языковые особенности англоязычных текстов технической документации / А. Ю. Горбунов, Л. А. Долбунова. – Текст : непосредственный // Огарёв-Online. – 2015. – № 4. – С. 131–138.

31. Данилина, И. И. Преподавание языков и технологий программирования в техническом вузе / И. И. Данилина. – Текст : непосредственный // Наука и образование в контексте глобальной трансформации: сборник статей XIII Международной научно-практической конференции (15 января 2024 г.). – Петрозаводск : МЦНП «НОВАЯ НАУКА», 2024. – С. 26–31.

32. Двуреченский, И. О. Веб-приложения: основы, технологии и разработка / И. О. Двуреченский, И. Н. Симонов, Л. В. Гаев. – Текст : непосредственный // Инновационная наука. – Липецк : Липецкий государственный технический университет, 2023. – С. 35–37.

33. Демьянков, В. З. Когнитивные техники трансфера знаний / В. З. Демьянков. – Текст : непосредственный // Когнитивные исследования языка. – 2016. – Вып. 26. – С. 29–32.

34. Дукаев, М. Ш. Использование искусственного интеллекта в автоматизации разработки веб-приложений / М. Ш. Дукаев, Е. В. Саровойтова, З. Б. Батчаева. – Текст : непосредственный // Современные наукоемкие технологии. Региональное приложение. – Грозный : ГГНТУ им. акад. Миллионщикова, 2024. – С. 91–96.

35. Ермаков, А. С. Перспективное развитие методологии DevOps / А. С. Ермаков. – Текст : непосредственный // Вестник НГУЭУ. – Новосибирск : Федеральное государственное бюджетное образовательное учреждение высшего образования «Новосибирский государственный университет экономики и управления «НИНХ», 2020. – С. 89–102.

36. Ершов, Т. А. Обзор современных технологий разработки для WEB-приложений / Т. А. Ершов, А. А. Голубничий. – Текст : непосредственный // Информационные технологии. – Москва : E-scio, 2023. – С. 172–177.

37. Жидков, А. В. Переводческие ошибки в научно-техническом переводе / А. В. Жидков. – Текст : непосредственный // Science Time. – 2014. – № 4. – С. 95–98.

38. Забело, Т. С. Обзор методологий разработки программного обеспечения: Agile, DevOps, Continuous Integration и другие подходы / Т. С. Забело, Е. И. Полоско. – Текст : непосредственный // Актуальные вопросы экономики и информационных технологий: материалы 60-й научной конференции аспирантов, магистрантов и студентов. – Минск : БГУИР, 2024. – С. 251–253.

39. Заварзин, В. И. Трансфер знаний и технологий / В. И. Заварзин. – Текст : непосредственный // Инженерный журнал: наука и инновации. – 2013. – С. 1–5.

40. Злобин, Е. А. Обзор препроцессоров языка каскадных таблиц стилей CSS / Е. А. Злобин, Т. В. Скворцова, В. И. Анциферова, Е. В. Грошева. – Текст : непосредственный // Перспективные аспекты моделирования систем и процессов. – Ростов-на-Дону : Молодой исследователь Дона, 2023. – № 30. – С. 56–63.

41. Золян, С. Т. К основаниям теории междисциплинарного трансфера. Принцип переводоведческой относительности / С. Т. Золян. – Текст : непосредственный // МЕТОД: Московский ежегодник трудов из обществоведческих дисциплин. – Москва : Федеральное государственное бюджетное учреждение науки «Институт научной информации по общественным наукам Российской академии наук», 2019. – С. 193–232.

42. Иванов, А. С. Разработка веб-приложений с использованием HTML, CSS и JavaScript / А. С. Иванов. – Москва : Издательство «Техносфера», 2018. – 1120 с. – Текст : непосредственный.

43. Калабина, Е. Г. Факторы обмена знаниями в процессе управления кросс-функциональными командами / Е. Г. Калабина, О. Ю. Беляк. – Текст : непосредственный. – 2021. – № 19 (4). – С. 515–547.

44. Калинина, С. С. Особенности научно-технической разновидности научного стиля в русском языке / С. С. Калинина. – Текст : непосредственный //

Вестник науки и образования. – Москва : Академия естествознания, 2017. – № 2 (26). – С. 65–69.

45. Карпенко, Л. Б. Проблемы цифровой межкультурной коммуникации на славянских языках / Л. Б. Карпенко. – Текст : непосредственный // Русин. – 2025. – № 80. – С. 269–285.

46. Клушина, Н. И. Жанровый аспект интернет-коммуникации / Н. И. Клушина. – Текст : непосредственный // Мултимедијална стилистика. – Бањалука, 2018. – Т. 1. – С. 61–68.

47. Клушина, Н. И. Дискурс-анализ и стилистика: интегративные методы исследования медиакоммуникации / Н. И. Клушина. – Текст : непосредственный // Вестник Российского университета дружбы народов. Серия: Лингвистика. – 2016. – Т. 20, № 4. – С. 78–90.

48. Княжева, Е. А. Оценка качества перевода: проблемы теории и практики / Е. А. Княжева. – Текст : непосредственный // Вестник ВГУ. Серия: Лингвистика и межкультурная коммуникация. – 2010. – № 2. – С. 190–195.

49. Комиссаров, В. Н. Теория перевода (лингвистические аспекты) / В. Н. Комиссаров. – Москва : Высшая школа, 1990. – 253 с. – Текст : непосредственный.

50. Коняева, Л. А. О некоторых трудностях научно-технического перевода / Л. А. Коняева. – Текст : непосредственный // Перевод и сопоставительная лингвистика. – Екатеринбург : Уральский гуманитарный институт, 2015. – № 11. – С. 50–54.

51. Котова, Ю. С. Особенности технического перевода / Ю. С. Котова. – Текст : непосредственный // Язык и культура (Новосибирск). – Новосибирск : Общество с ограниченной ответственностью «Центр развития научного сотрудничества», 2013. – С. 133–138.

52. Куваев, М. Ю. Современные тенденции развития веб-разработки / М. Ю. Куваев, О. В. Антимонов. – Текст : непосредственный // Научно-образовательный

журнал для студентов и преподавателей «StudNet». – Москва : ООО «Электронная наука», 2020. – № 9. – С. 1269–1273.

53. Кужугет, Ш. Ю. Проблемы перевода концептов культуры на другой язык: на примере тувинских концептов культуры / Ш. Ю. Кужугет, Н. Д. Сувандии, Ч. К. Ламажаа. – Текст : непосредственный // Полилингвиальность и транскультурные практики. – Москва : Федеральное государственное автономное образовательное учреждение высшего образования «Российский университет дружбы народов», 2021. – С. 405–420.

54. Латышев, Л. К. Современные проблемы общей теории перевода / Л. К. Латышев, Н. Ю. Северова. – Текст : непосредственный // Вестник Брянского государственного университета. – 2017. – № 1. – С. 297–301.

55. Латышев, Л. К. Технология перевода / Л. К. Латышев. – 2-е изд., перераб. и доп. – Москва : Издательский центр «Академия», 2005. – 320 с. – Текст : непосредственный.

56. Лотман, Ю. М. Культура и взрыв / Ю. М. Лотман. – Москва : Прогресс, Гнозис, 1992. – 272 с. – Текст : непосредственный.

57. Лю, Лифэнь. Анализ прагмалингвистических неудач в русском переводе знаков для общественных мест / Лифэнь Лю, Сяохун У. – Текст : непосредственный // Вестник Санкт-Петербургского университета. Востоковедение и африканистика. – 2025. – Т. 17, № 2. – С. 287–302.

58. Люсый, А. П. Перевод как культурная практика рефлексии, в частности философской / А. П. Люсый. – Текст : непосредственный // Вестник Московского государственного лингвистического университета. Общественные науки. – 2016. – № 2 (741). – С. 75–82.

59. Матвеев, В. С. Современные фреймворки для разработки информационных систем на примере платформы Vue / В. С. Матвеев. – Текст : непосредственный // Форум молодых ученых. – 2020. – № 5. – С. 301–306.

60. Матвеев, Н. А. Когнитивные и лингвокультурные аспекты переводческих неудач / Н. А. Матвеев, Г. А. Шушарина. – Текст : непосредственный // European Journal of Education and Applied Psychology. – 2015. – № 5. – С. 45.

61. Матвеева, А. В. WEB-разработки / А. В. Матвеева. – Текст : непосредственный // Международный журнал прикладных наук и технологий «Integral». – 2019. – № 3. – Москва : Общество с ограниченной ответственностью «Электронная наука». – С. 535–539.

62. Мишкуров, Э. Н. Герменевтико-переводческий методологический стандарт в зеркале трансдисциплинарности (часть 1) / Э. Н. Мишкуров. – Текст : непосредственный // Вестник Московского университета. Серия 22. Теория перевода. – 2015. – № 2. – С. 17–37.

63. Нефёдов, Д. С. Обоснование выбора React для разработки веб-приложения для тестирования в образовательных организациях: сравнение с нативным JavaScript / Д. С. Нефёдов, Д. Р. Шехирев. – Текст : непосредственный // Парадигма. – 2025. – № 4. – С. 315–321.

64. Нишонов, У. И. Особенности перевода технических текстов / У. И. Нишонов. – Текст : непосредственный // Вестник науки и образования. – Москва : Академия естествознания, 2020. – № 14 (92). – С. 46–48.

65. Оберюхтина, М. В. Когнитивные стратегии в переводе пиар-дискурса в романе К. Бакли «День бумеранга» / М. В. Оберюхтина. – Текст : непосредственный // Филология и человек. – Барнаул : Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный университет», 2022. – С. 151–157.

66. Перевод MDN на русский язык #276. – Текст : электронный // GitHub. – URL: https://github.com/mdn/translated-content/issues/276 (дата обращения: 03.12.2025).

67. Петров, В. Н. Веб-разработка: основы и современные технологии / В. Н. Петров. – СПб. : БХВ-Петербург, 2019. – 760 с. – Текст : непосредственный.

68. Пильщиков, И. А. Неполная переводимость как механизм познания и коммуникации / И. А. Пильщиков. – Текст : непосредственный // Лингвистика и семиотика межкультурных трансферов. Методы, принципы, технологии. – Москва : Культурная революция, 2016. – С. 203–233.

69. Полетаева, Е. Д. К вопросу о переводе англоязычных технических терминов на русский язык / Е. Д. Полетаева, О. О. Сосновская. – Текст : непосредственный // Огарёв-Online. – Саранск : Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский Мордовский государственный университет им. Н. П. Огарёва», 2014. – 7 с.

70. Поликарпов, А. М. Интегративное переводоведение: предпосылки возникновения и основные идеи / А. М. Поликарпов. – Текст : непосредственный // Вестник Волгоградского государственного университета. Серия 2: Языкознание. – 2017. – Т. 16, № 3. – С. 6–17.

71. Поликарпов, А. М. Переводческое стратегирование: сознательное и бессознательное / А. М. Поликарпов. – Текст : непосредственный // Новое и традиционное в переводоведении и преподавании русского языка как иностранного. – Панъевропейский университет «АПЕИРОН», Баня-Лука, 2025. – С. 216–225.

72. Поликарпов, А. М. Передача охотничьей лексики при переводе произведений северных авторов на немецкий язык / А. М. Поликарпов. – Текст : непосредственный // Вестник Сыктывкарского университета. Серия гуманитарных наук. – 2021. – № 1 (17). – С. 43–56.

73. Поликарпов, А. М. Синтаксическая полифункциональность / А. М. Поликарпов. – Архангельск : Северный (Арктический) Федеральный Университет имени М. В. Ломоносова, 2014. – 214 с. – Текст : непосредственный.

74. Поликарпов, А. М. Переводческое мышление с позиций интегративного переводоведения / А. М. Поликарпов // Перевод, переводоведение и дидактика

перевода. Северные регионы России : коллективная монография / под общ. ред. Н. Н. Гавриленко. – М. : Флинта, 2026. – С. 117–186. – Текст : непосредственный.

75. Полякова, С. Е. Понятие и причины возникновения коммуникативных неудач / С. Е. Полякова. – Текст : непосредственный // Вестник Ленинградского государственного университета им. А. С. Пушкина. – 2012. – Т. 7, № 1. – С. 83–92.

76. Прияцелюк, Н. Веб-фреймворки для начинающих: простое объяснение с примерами / Н. Прияцелюк. – Текст : электронный. – URL: https://tproger.ru/translations/web-frameworks-how-to-get-started/ (дата обращения: 05.06.2025).

77. Пумпянский, А. Л. Введение в практику перевода научной и технической литературы на английский язык / А. Л. Пумпянский. – Москва : Наука, 1965. – 304 с. – Текст : непосредственный.

78. Распокина, Л. П. Эрратология перевода / Л. П. Распокина. – Текст : непосредственный // Теория перевода в классической и современной исследовательской парадигме. Онтология, методология, аксиология. – Москва : Флинта. – С. 257–270.

79. Ратушняк, Е. А. Исследование производительности современных клиентских веб-фреймворков / Е. А. Ратушняк. – Текст : непосредственный // Программные системы и вычислительные методы. – 2025. – № 2. – С. 63–78.

80. Рецкер, Я. И. Методика технического перевода / Я. И. Рецкер. – Москва : Дрофа, 2009. – 216 с. – Текст : непосредственный.

81. Руденко, Е. С. От переводческой ошибки к переводческой стратегии / Е. С. Руденко, З. Б. Темирханова, Р. И. Исаева. – Текст : непосредственный // Вестник Московского государственного областного университета. Серия: Лингвистика. – 2021. – № 1. – С. 129–142.

82. Сазонова, В. М. Трансфер знаний между наукой и бизнесом: барьеры, мотивации и перспективы сотрудничества / В. М. Сазонова. – Текст : непосредственный // Инновационная наука. – 2025. – № 5-2. – С. 127–130.

83. Сдобников, В. В. Коммуникативно-функциональный подход в переводоведении / В. В. Сдобников. – Текст : непосредственный // Казанский вестник молодых ученых. – Казань : Казанский федеральный университет, 2016.

84. Сдобников, В. В. Новые тенденции в переводоведении / В. В. Сдобников. – Текст : непосредственный // Казанский вестник молодых учёных. – Казань : Казанский федеральный университет, 2018. – С. 56–74.

85. Сдобников, В. В. Оценка качества перевода (коммуникативно-функциональный подход): монография / В. В. Сдобников. – Москва : ФЛИНТА : Наука, 2015. – 112 с. – Текст : непосредственный.

86. Сдобников, В. В. Оценка качества перевода в аудитории и в реальной жизни / В. В. Сдобников. – Текст : непосредственный // Перевод и сопоставительная лингвистика. – 2015. – Вып. 11. – С. 41–45.

87. Сдобников, В. В. Оценка качества перевода: коммуникативно-функциональный подход / В. В. Сдобников. – Текст : непосредственный // Социальные и гуманитарные науки. Отечественная и зарубежная литература. Сер. 6, Языкознание: реферативный журнал. – Москва : Флинта : Наука, 2015. – 112 с.

88. Семёнов, Д. В. Классификация руководств пользователя как разновидность технического текста / Д. В. Семёнов. – Текст : непосредственный // Международный научный журнал «Вестник науки». – Таганрог, 2023. – № 6 (63). – С. 809–814.

89. Сергачева, М. А. Анализ фреймфорков для разработки современных веб-приложений / М. А. Сергачева, К. А. Михалевская. – Текст : непосредственный // Кронос: естественные и технические науки. – Н. Новгород : Serenity-Group, 2020. – С. 35–39.

90. Сидорова, Е. А. Семантические особенности технической документации: онтологический взгляд на проблемы корректности / Е. А. Сидорова, Н. О. Гаранина, И. С. Кононенко, О. И. Боровикова. – Текст : непосредственный // Информационные

и математические технологии в науке и управлении. – Москва, 2018. – № 4 (12). – С. 30–41.

91. Смирнова, Т. Е. Адекватность перевода в коммуникативном аспекте / Т. Е. Смирнова, В. Е. Ермишина. – Текст : непосредственный // Огарёв-Online. – 2017. – № 14 (103). – С. 1–5.

92. Соколова, Н. В. Адекватность и эквивалентность как основные критерии оценки качества перевода в зарубежном и отечественном переводоведении / Н. В. Соколова. – Текст : непосредственный // Теория перевода в классической и современной исследовательской парадигме. Онтология, методология, аксиология. – Москва : Флинта. – С. 230–256.

93. Софронова, Т. М. Формирование прагматической компетенции студента-переводчика в свете теории ошибок и переводческих несоответствий / Т. М. Софронова, Т. П. Бабак. – Текст : непосредственный // Вестник Красноярского государственного педагогического университета им. В. П. Астафьева. – 2022. – № 3. – С. 15–28.

94. Станиславский, А. Р. Типология научно-технических текстов и перевод: от традиционных моделей до моделей в рамках концепции LSP / А. Р. Станиславский. – Текст : электронный // Гуманитарные научные исследования. – 2016. – № 8. – URL: https://human.snauka.ru/2016/08/15254 (дата обращения: 11.06.2025).

95. Ступина, М. В. Использование облачных интернет-технологий в разработке веб-ориентированных информационных систем / М. В. Ступина. – Текст : непосредственный // Инженерный вестник Дона. – Ростов-на-Дону : Донской государственный технический университет, 2023. – С. 130–139.

96. Суховаров, И. В. Трансфер знаний в образовательном кластере как механизм формирования ключевых компетенций в вузе / И. В. Суховаров. – Текст : непосредственный // Развитие экономики и менеджмента в современном мире:

сборник научных трудов по итогам международной конференции. – Курск : Курский государственный университет, 2016. – С. 45–52.

97. Терехина, С. В. Управление знаниями современной организации в условиях цифровизации / С. В. Терехина. – Текст : непосредственный // Инновационная экономика: перспективы развития и совершенствования. – Курск : Закрытое акционерное общество «Университетская книга», 2023. – № 3 (69). – С. 158–163.

98. Тиллаева, Ш. М. Основные особенности технического перевода / Ш. М. Тиллаева. – Текст : непосредственный // Достижения науки и образования. – Москва : ООО «Олимп», 2019. – С. 15–16.

99. Узюмова, Н. В. Адаптация новых сотрудников как критерий социальной эффективности корпоративной культуры организации / Н. В. Узюмова. – Текст : непосредственный // Гуманитарные науки. Вестник Финансового университета. – 2019. – № 3 (39). – С. 31–34.

100. Федоров, В. В. Новые информационные технологии и некоторые особенности их практической реализации / В. В. Федоров. – Текст : непосредственный // Вестник Российской таможенной академии. – 2008. – № 4 (5). – Люберцы : Государственное казенное образовательное учреждение высшего профессионального образования. – С. 48–51.

101. Федюченко, Л. Г. Технический текст как объект перевода / Л. Г. Федюченко. – Текст : непосредственный // Филологические науки. Вопросы теории и практики. – Тамбов : Грамота, 2019. – № 7. – С. 292–296.

102. Флек, М. Б. Факторы, проблемы и перспективы трансфера знаний в организации высокотехнологичного сектора / М. Б. Флек, Е. А. Угнич. – Текст : непосредственный // Экономические и социальные перемены: факты, тенденции, прогноз. – Вологда : Федеральное государственное бюджетное учреждение науки «Институт социально-экономического развития территорий Российской академии наук», 2023. – С. 83–100.

103. Хайруллин, Д. А. Изучение использования typescript в react-приложениях, преимущества статической типизации, подходы к типизации компонентов, хуков и состояния / Д. А. Хайруллин, М. Р. Богданов. – Текст : непосредственный // Международный журнал гуманитарных и естественных наук. – 2024. – С. 233–238.

104. Цвиллинг, М. Я. Исчерпала ли теория перевода свои возможности? / М. Я. Цвиллинг. – Текст : непосредственный // Вопросы филологии. – 2002. – № 1. – С. 48–50.

105. Цибизова, Т. Ю. «Менторство» как элемент методики работы преподавателя в системе высшего профессионального образования / Т. Ю. Цибизова, О. М. Августан, Д. А. Сергеев, С. А. Марданов. – Текст : непосредственный // Современные проблемы науки и образования. – 2017. – № 4. – Москва : ООО «Издательский дом Академия естествознания». – С. 141.

106. Шаповалова, А. Ф. Методология оценки качества перевода в обучении и переводческой практике / А. Ф. Шаповалова. – Текст : непосредственный // Языковая личность и эффективная коммуникация в современном поликультурном мире: материалы V Международной научно-практической конференции, посвящённой 20-летию основания каф. теории и практики перевода фак. социокультурных коммуникаций Белорусского государственного университета. – Минск, 2019. – С. 305–311.

107. Швейцер, А. Д. Теория перевода: статус, проблемы, аспекты / А. Д. Швейцер. – Москва : Наука, 1988. – 215 с. – Текст : непосредственный.

108. Шутаева, Е. А. Трансфер знаний и технологий как важнейшее направление формирования «новой экономики» европейских стран / Е. А. Шутаева. – Текст : непосредственный // Наука и мир. – 2015. – № 8. – С. 90–92.

109. Щелокова, А. А. Эрратология как общая теория ошибок / А. А. Щелокова. – Текст : непосредственный // Социо- и психолингвистические исследования. – 2020. – № 8. – С. 168–172.

110. Щипицина, Л. Ю. Компьютерно-опосредованная коммуникация: лингвистический аспект анализа / Л. Ю. Щипицина. – М. : КРАСАНД, 2010. – 296 с. – Текст : непосредственный.

111. Якименко, А. А. Разработка платформы для управления инфраструктурой интернета вещей / А. А. Якименко, А. И. Белов, П. С. Гончарук, И. М. Стубарев. – Текст : непосредственный // Известия Самарского научного центра Российской академии наук. – 2017. – С. 97–105.

112. Яковлев, А. А. К вопросу об эрратологии в переводе / А. А. Яковлев. – Текст : непосредственный // Lingua mobilis. – 2010. – № 5. – С. 111–116.

113. Яровая, Е. В. Современные инструменты для написания веб-приложений / Е. В. Яровая. – Текст : непосредственный // Столыпинский вестник. – 2022. – № 5. – С. 2782–2786.

114. Adler, C. Conceptualizing the transfer of knowledge across cases in transdisciplinary research / C. Adler, G. Hirsch Hadorn, T. Breu, U. Wiesmann, C. Pohl. – Text : direct // Sustainability Science. – 2018. – Vol. 13. – P. 179–190.

115. Ali, Z. The History of React.js: From Facebook's Internal Tool to Global... / Z. Ali. – Текст : электронный. – URL: https://zeeshan.p2pclouds.net/blogs/history-of-reactjs/ (дата обращения: 19.09.2025).

116. Allen, J. React example viability for efficient API learning (REVEAL): a tool to help programmers utilize incompatible code examples in React.js / J. Allen, C. Kelleher. – Text : direct // Journal of Computer Languages. – 2023. – Vol. 75. – Art. 101201.

117. Angular, заглавная страница. – Текст : электронный. – 2024. – URL: https://angular.dev/overview (дата обращения: 10.06.2025).

118. Angular: The Web Development Framework for Building Modern Apps. – Текст : электронный. – URL: https://angular.io (дата обращения: 05.06.2025).

119. Aouni, F. A systematic literature review on Agile, Cloud, and DevOps integration: Challenges, benefits / F. Aouni, K. Moumane, A. Idri, M. Najib, S. U. Jan. – Text : direct // Information and Software Technology. – 2025. – Vol. 177. – Art. 107569.

120. Argote, L. Knowledge Transfer within organizations: Mechanisms, Motivation, and Consideration / L. Argote. – Text : direct // Annual Reviews of Psychology. – 2023. – P. 405–431.

121. Argote, L. Knowledge Transfer: A Basis for Competitive Advantage in Firms / L. Argote, P. Ingram. – Text : direct // Organizational Behavior and Human Decision Processes. – 2000. – No. 82. – P. 150–169.

122. Bacon, E. On the Combinatory Nature of Knowledge Transfer Conditions: A Mixed Method Assessment / E. Bacon, M. D. Williams, G. H. Davies. – Text : direct // Information Systems Frontier. – 2023. – Vol. 25. – P. 1039–1061.

123. Bezverhiy, O. Optimization of cross-platform applications using the React library / O. Bezverhiy, O. Kutsenko. – Text : direct // World Science. – № 3 (85). – P. 1–5.

124. Blanco, G. Understanding the social evolution of the Java community in Stack Overflow: A 10-year study of developer interactions / G. Blanco, R. Pérez-López, F. Fdez-Riverola, A. M. G. Lourenço. – Text : direct // Future Generation Computer Systems. – 2020. – Vol. 105. – P. 446–454.

125. Bogner, J. Do RESTful API design rules have an impact on the understandability of Web APIs? / J. Bogner, S. Kotstein, T. Pfaff. – Text : direct // Empirical Software Engineering. – 2023. – Vol. 28, Iss. 6. – Art. 132.

126. Chambers 20th Century Dictionary. – Edinburgh : Chambers, 1983. – 1583 p. – Text : direct.

127. D3.js, онлайн-руководство по применению. – Текст : электронный. – 2024. – URL: https://observablehq.com/plot/ (дата обращения: 10.06.2025).

128. Dolmans, S. A. M. Knowledge transfer to industry: how academic researchers learn to become boundary spanners during academic engagement / S. A. M. Dolmans, B. Walrave, S. Read, N. van Stijn. – Text : direct // Journal of Technology Transfer. – 2022. – Vol. 47, Iss. 5. – P. 1422–1450.

129. Dybå, T. Empirical studies of agile software development: A systematic review / T. Dybå, T. Dingsøyr. – Text : direct // Information and Software Technology. – 2008. – Vol. 50, Iss. 9–10. – P. 833–859.

130. Emmanni, P. S. Comparative Analysis of Angular, React, and Vue.js in Single Page Application Development / P. S. Emmanni. – Text : direct // International Journal of Science and Research. – 2023. – Vol. 12, Iss. 6. – P. 2971–2974.

131. Eoyang, E. C. Translating as a mode of thinking translation as a model of thought / E. C. Eoyang. – Text : direct // Perspectives: Studies in Translation Theory and Practice. – 1996. – Vol. 4, Iss. 1. – P. 53–69.

132. Fan, Q. Why API documentation is insufficient for developers: an empirical study / Q. Fan, Y. Yu, T. Wang, G. Yin, H. Wang. – Text : direct // Science China Information Sciences. – 2021. – Vol. 64. – Art. 119102.

133. Ferrer-Serrano, M. Knowledge Transfer and Networks: A Bibliometric Approach Through Performance Analysis, Science mapping, and Dynamic Network Analysis / M. Ferrer-Serrano, L. Fuentelsaz, M. P. Latorre-Martínez. – Text : direct // Journal of the Knowledge Economy. – 2026. – Vol. 17. – P. 5237–5272.

134. Fregnan, E. Graph-based visualization of merge requests for code review / E. Fregnan, J. Fröhlich, D. Spadini, A. Bacchelli. – Text : direct // Journal of Systems and Software. – 2023. – Vol. 195. – Art. 111506.

135. Garifulin, R. Development and implementation of high-speed frontend architectures for complex enterprise systems / R. Garifulin. – Text : direct // Холодная наука. – 2024. – № 12. – С. 56–63.

136. Gili, Y. S. S. An Analysis of Translation Error Made by Students at The Sixth Semester of English Department in IKIP Muhammadiyah Maumere / Y. S. S. Gili, N. S. Ekhsan. – Text : direct // Udayana Journal of Social Sciences and Humanities. – 2023. – № 7 (2). – P. 1–10.

137. González-Mora, C. Improving open data web API documentation through interactivity and natural language generation / C. González-Mora, C. Barros, I. Garrigós, J.

Zubcoff, E. Lloret, J.-N. Mazón. – Text : direct // Computer Standards & Interfaces. – 2023. – Vol. 83. – Art. 103657.

138. Hamori, F. The History of React.js on a Timeline - RisingStack Engineering / F. Hamori. – Текст : электронный. – URL: https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/ (дата обращения: 18.09.2025).

139. Hansen, G. Translation 'errors' / G. Hansen. – Текст : электронный. – 2015. – Copenhagen Business School [Электронный ресурс]. – URL: https://benjamins.com/online/hts/articles/tra3 (дата обращения: 08.12.2025).

140. Herfeld, C. Knowledge transfer and its contexts / C. Herfeld, C. Lisciandra. – Text : direct // Studies in History and Philosophy of Science. – 2019. – Vol. 77. – P. 1–10.

141. History and Evolution of React - GeeksforGeeks. – Текст : электронный. – URL: https://www.geeksforgeeks.org/reactjs/history-and-evolution-of-react/ (дата обращения: 18.09.2025).

142. Hoffman, D. API documentation with executable examples / D. Hoffman, P. Strooper. – Text : direct // Journal of Systems and Software. – 2003. – Vol. 66, Iss. 2. – P. 143–156.

143. Iffland, D. Microsoft, Google, and Mozilla Team Up for Web Documentation / D. Iffland. – Текст : электронный // InfoQ. – 2017. – URL: https://www.infoq.com/news/2017/10/microsoft-google-mozilla-mdn (дата обращения: 03.12.2025).

144. Kniazev, I. Exploring the frontend industry in 2022: A comprehensive analysis of popular technologies, tools, and development practices / I. Kniazev. – Text : direct // Sciences of Europe. – 2023. – № 110. – С. 64–77.

145. Komperla, V. React: A detailed survey / V. Komperla, P. Deenadhayalan, P. Ghuli, R. Pattar. – Text : direct // Indonesian Journal of Electrical Engineering and Computer Science. – 2022. – Vol. 26, Iss. 3. – P. 1710–1717.

146. Kotstein, S. RESTBERTa: a Transformer-based question answering approach for semantic search in Web API documentation / S. Kotstein, C. Decker. – Text : direct // Cluster Computing. – 2024. – Vol. 27, Iss. 4. – P. 4035–4061.

147. Li, X. A novel integrated approach for quantifying the convergence of disruptive technologies from science to technology / X. Li, Y. Wang. – Text : direct // Technological Forecasting and Social Change. – 2024. – Vol. 209. – Art. 123825.

148. Lotman, Yu. M. Semiosfera [Semiosphere] / Yu. M. Lotman. – Moscow, 2000. – Text : direct.

149. Lundmark, C. Taking Stock of Knowledge Transfer Studies: Finding Ways Forward / C. Lundmark, J. Nilsson, A. Krook-Riekkola. – Text : direct // Environmental Management. – 2023. – Vol. 72. – P. 1146–1162.

150. Marchiori, D. Knowledge transfer in the context of inter-organizational networks: Foundations and intellectual structures / D. Marchiori, M. Franco. – Text : direct // Journal of Innovation & Knowledge. – 2020. – Vol. 5, Iss. 2. – P. 130–139.

151. Massaro, M. Trust, control and knowledge transfer in small business networks / M. Massaro, A. Moro, E. Aschauer, M. Fink. – Text : direct // Review of Managerial Science. – 2019. – Vol. 13, Iss. 2. – P. 267–301.

152. McCarthy, K. S. The Multidimensional Knowledge in Text Comprehension framework / K. S. McCarthy, D. S. McNamara. – Text : direct // Educational Psychologist. – 2021. – Vol. 56, Iss. 3. – P. 196–214.

153. MDN Web Docs. Изучение веб-разработки. – Текст : электронный. – URL: https://developer.mozilla.org/ru/docs/Learn_web_development (дата обращения: 05.06.2025).

154. Meng, M. How Developers Use API Documentation: An Observation Study / M. Meng, S. Steinhardt, A. Schubert. – Text : direct // Communication Design Quarterly. – 2019. – Vol. 7, Iss. 2. – P. 40–49.

155. Morris, C. W. Writings on the General Theory of Signs / C. W. Morris. – The Hague ; Paris : Mouton, 1971. – 486 p. – Text : direct.

156. Myllärniemi, V. Development as a journey: factors supporting the adoption and use of software frameworks / V. Myllärniemi, S. Kujala, M. Raatikainen, P. Sevón. – Text : direct // Journal of Software Engineering Research and Development. – 2018. – Vol. 6. – Art. 6.

157. Ngoc, N. T. N. An Analysis on Translation Errors in Online Vietnamese-English Menus / N. T. N. Ngoc, N. Le, N. L. M. Phương, L. A. Khoa, P. D. Khoa, T. T. T. Lam. – Text : direct // 18th Internationall Coference of the Asia Association of Computer - Assisgeeted. – 2021. – P. 299–310.

158. Nidhra, S. Knowledge transfer challenges and mitigation strategies in global software development: A systematic literature review and industrial validation / S. Nidhra, M. Yanamadala, W. Afzal, R. Torkar. – Text : direct // International Journal of Information Management. – 2013. – Vol. 33, Iss. 2. – P. 333–355.

159. Niu, M. The Development of Pragmatics in Morris's Behavioral Semiotics: Semiotic Perspective / M. Niu, T. Dechsubha. – Text : direct // Education Quarterly Reviews. – 2022. – Vol. 5, Iss. 2. – P. 216–225.

160. Orazgeldiyeva, N. Integration of Artificial Intelligence with React JS web framework / N. Orazgeldiyeva, A. Babayev, A. Akmyradov, S. Jemhurov. – Text : direct // Наука и мировоззрение. – Балканабат : Научное издательство «Наука и мировоззрение», 2025. – № 1 (36). – P. 293–297.

161. Paramadani, R. B. User Interface Rendering Time in Android Applications: Revealing the Effects of Design Patterns / R. B. Paramadani, M. A. Akbar, A. Pinandito. – Text : direct // SIET 2023: Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology. – New York : ACM, 2023. – P. 631–636.

162. Patel, D. Explore the Evolution of React: A Detailed React Version History / D. Patel. – Текст : электронный. – URL: https://www.dhiwise.com/post/react-version-history-the-evolution-of-a-javascript-powerhouse (дата обращения: 18.09.2025).

163. Prieto Ramos, F. Quality Assurance in Legal Translation: Evaluating Process, Competence and Product in the Pursuit of Adequacy / F. Prieto Ramos. – Text : direct // International Journal for the Semiotics of Law. – 2015. – Vol. 28, Iss. 1. – P. 11–30.

164. React: The Library for Web and Native User Interfaces. – Текст : электронный. – URL: https://react.dev (дата обращения: 05.06.2025).

165. Rodić, M. On the impersonal stance in scientific writing: A cross-disciplinary and cross-cultural study / M. Rodić. – Text : direct // TEME. – 2025. – Vol. 49, Iss. 4. – P. 805–820.

166. Schafer, I. C. Mozilla leads project to consolidate web development documentation across browsers / I. C. Schafer. – Текст : электронный // SD Times. – 2017. – URL: https://sdtimes.com/google/mozilla-leads-project-consolidate-web-development-documentation-across-browsers (дата обращения: 04.12.2025).

167. Shogenov, A. M. Advantages and possibilities of ReactJS in modern web development / A. M. Shogenov. – Text : direct // Восточно-европейский научный журнал. – Санкт-Петербург : Общество с ограниченной ответственностью «Логика+», 2023. – № 8-1 (93). – С. 36–40.

168. Sidorov, D. A comparative analysis of component-based architectures in web design for scalable applications / D. Sidorov. – Text : direct // Холодная наука. – 2024. – № 8. – С. 24–31.

169. Storey, V. C. Generative Artificial Intelligence: Evolving Technology, Growing Societal Impact, and Opportunities for Information Systems Research / V. C. Storey, W. T. Yue, J. L. Zhao, R. Lukyanenko. – Text : direct // Information Systems Frontiers. – 2025. – Vol. 27, Iss. 5. – P. 2081–2102.

170. Sulistiyaningsih, E. F. Investigating Translation Errors in Recount Texts: A Study of Midwifery Students' use of Machine Translation / E. F. Sulistiyaningsih, Z. Wafa, E. Susilowati. – Text : direct // English Education and Literature Journal. – 2025. – Vol. 5, Iss. 02. – P. 346–356.

171. Tabares, R. HTML5 and the evolution of HTML; tracing the origins of digital platforms / R. Tabares. – Text : direct // ScienceDirect. – Elsevier, 2021. – № 65. – С. 124–129.

172. Taylor, C. The History of ReactJS - HRTech Cube / C. Taylor. – Текст : электронный. – URL: https://hrtechcube.com/the-history-of-reactjs/ (дата обращения: 19.09.2025).

173. Temizkan, O. Knowledge transfer to aid social coding: The case of Stack Overflow / O. Temizkan, R. L. Kumar. – Text : direct // Decision Support Systems. – 2024. – Vol. 179.

174. Tien, N. H. Knowledge transfer and succession process in small family businesses / N. H. Tien, N. Trang. – Text : direct // Int. J. Entrepreneurship and Small Business. – 2023. – P. 34–51.

175. Toury, G. Descriptive Translation Studies – and Beyond / G. Toury. – Москва : John Benjamins Publishing Co., 2012. – 311 p. – Текст : непосредственный.

176. Uddin, G. Towards crowd-sourced API documentation / G. Uddin, F. Khomh, C. K. Roy. – Text : direct // 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings. – 2019. – P. 310–311.

177. Ward, V. L. Knowledge brokering: Exploring the process of transferring knowledge into action / V. L. Ward, A. O. House, S. Hamer. – Text : direct // BMC Health Services Research. – 2009. – Vol. 9. – Art. 12.

178. Web Design Museum. React. – Текст : электронный. – URL: https://www.webdesignmuseum.org/web-design-history/react-2013 (дата обращения: 02.12.2025).

179. Why Choose React For Web Development in 2025 - GeeksforGeeks. – Текст : электронный. – URL: https://www.geeksforgeeks.org/blogs/why-choose-react-for-web-development/ (дата обращения: 20.09.2025).

180. Xu, J. A philosophical understanding of translation errors: An examination of subjective and objective mechanisms / J. Xu. – Text : direct // InContext. – 2025. – Vol. 5, Iss. 2. – P. 108–133.

181. Zaidman, N. Knowledge Transfer Within Multinationals and Their Foreign Subsidiaries: A Culture-Context Approach / N. Zaidman, D. M. Brock. – Text : direct // Group & Organization Management. – 2009. – Vol. 34, Iss. 3. – P. 297–329.

182. Zolkifli, N. N. Version Control System: A Review / N. N. Zolkifli, A. Ngah, A. Deraman. – Text : direct // Procedia Computer Science. – 2018. – Vol. 135. – P. 408–415.

Словари и справочники

183. Большой толковый словарь русского языка / Сост. и гл. ред. С. А. Кузнецов. – Санкт-Петербург : Норинт, 2000. – 1536 с. – Текст : непосредственный.

184. Нелюбин, Л. Л. Толковый переводоведческий словарь / Л. Л. Нелюбин. – 3-е изд., перераб. – М. : Флинта : Наука, 2003. – 320 с. – Текст : непосредственный.

185. Основные понятия переводоведения (отечественный опыт). Терминологический словарь-справочник / отв. ред. М. Б. Раренко. – М. : ИНИОН РАН, 2010. – 260 с. – Текст : непосредственный.

Источники эмпирического материала

186. MDN Web Docs. – Текст : электронный. – URL: https://developer.mozilla.org/en-US/about (дата обращения: 03.12.2025).

187. MDN Web Docs. Русскоязычная версия. – Текст : электронный. – URL: https://developer.mozilla.org/ru/ (дата обращения: 03.12.2025).

188. React JS. Официальная документация. – Текст : электронный. – URL: https://legacy.reactjs.org/ (дата обращения: 22.09.2025).

189. React JS. Официальная документация. Русскоязычная версия. – Текст : электронный. – URL: https://ru.legacy.reactjs.org/ (дата обращения: 22.09.2025).

190. Vue.js Официальная документация. – Текст : электронный. – URL: https://vuejs.org/ (дата обращения: 25.11.2025).

191. Vue.js Официальная документация. Русскоязычная версия – Текст : электронный. – URL: https://ru.vuejs.org/ (дата обращения: 25.11.2025).