

**Кузьминов Андрей Петрович**

студент

**Решетняк Денис Андреевич**

студент

**Латышов Матвей Михайлович**

студент

*Научный руководитель*

**Прокончук Анна Реональдовна**

старший преподаватель

ФГБОУ ВО «МИРЭА – Российский технологический университет»

г. Москва

## **АНГЛОЯЗЫЧНЫЕ ЦИФРОВЫЕ РЕСУРСЫ**

### **В ПОДГОТОВКЕ РАЗРАБОТЧИКОВ КЛИЕНТ-СЕРВЕРНЫХ СИСТЕМ**

***Аннотация:** в статье рассматривается роль англоязычных цифровых ресурсов в профессиональной подготовке разработчиков клиент-серверных систем и в формировании иноязычной профессиональной компетенции студентов технологического вуза. Актуальность темы обусловлена тем, что подавляющая часть технической документации, стандартов и профессиональной коммуникации в области клиент-серверной разработки представлена на английском языке. Цель работы – систематизировать основные виды англоязычных цифровых ресурсов и определить педагогические условия их эффективного использования в учебном процессе. Предложена классификация ресурсов по функциональному назначению, проанализированы языковые трудности при работе с аутентичными материалами и официальными стандартами, а также сформулированы практический алгоритм и критерии оценивания такой работы.*

***Ключевые слова:** английский язык, клиент-серверные системы, цифровые ресурсы, профессиональная компетенция, иноязычная подготовка.*

*Введение.*

Современная разработка программного обеспечения носит выраженный международный характер, а английский язык фактически выступает основным языком профессиональной коммуникации в сфере информационных технологий. Особенно отчётливо это проявляется в области клиент-серверных систем – архитектурного подхода, при котором приложение разделено на клиентскую часть, формирующую запросы, и серверную часть, обрабатывающую эти запросы и возвращающую результат. Спецификации сетевых протоколов, документация фреймворков и описания программных интерфейсов создаются преимущественно на английском языке, поэтому иноязычная профессиональная компетенция становится непосредственным условием профессиональной состоятельности будущего разработчика [3].

Студент, не способный читать оригинальную техническую документацию, оказывается ограничен переводными и нередко устаревшими источниками. Вместе с тем цифровая среда предоставляет преподавателю иностранного языка богатый набор аутентичных ресурсов для одновременного развития языковых навыков и профессиональных знаний. Цель настоящей работы – рассмотреть англоязычные цифровые ресурсы как средство подготовки разработчиков клиент-серверных систем и определить педагогические условия их эффективного применения в учебном процессе.

### *Роль английского языка в разработке клиент-серверных систем.*

Клиент-серверная архитектура лежит в основе большинства современных веб-приложений, мобильных сервисов и распределённых систем. Взаимодействие между клиентом и сервером строится на основе протоколов и соглашений, базовая терминология которых сформировалась в англоязычной среде. Ключевые понятия – request, response, endpoint, status code, authentication, caching – не имеют устойчивых однословных русских эквивалентов и используются специалистами преимущественно в оригинальном виде. Общеизвестные учебные материалы, например описание модели клиент-серверного взаимодействия в справочном ресурсе MDN Web Docs, изначально публикуются на английском языке и лишь частично переводятся [8]. Это означает, что освоение предметной области

неотделимо от освоения соответствующего англоязычного терминологического аппарата.

Принципы построения программных интерфейсов, в частности архитектурный стиль REST (Representational State Transfer), изначально были описаны на английском языке, и наиболее полные их изложения по-прежнему доступны именно в оригинальных источниках [5]. Профессиональные стандарты этой области также формируются на английском языке: семантика протокола HTTP закреплена в спецификации RFC 9110, разработанной IETF (Internet Engineering Task Force) [6], описания программных интерфейсов API (Application Programming Interface) оформляются в соответствии со спецификацией OpenAPI [9], а типовые риски безопасности систематизированы в рекомендациях международного консорциума OWASP (Open Worldwide Application Security Project) [10]. Таким образом, для разработчика клиент-серверных систем чтение англоязычной документации является не дополнительной нагрузкой, а повседневной рабочей операцией. Это определяет специфику языковой подготовки: акцент смещается с общего владения языком на способность точно понимать технический текст и корректно использовать профессиональную лексику, что соответствует принципам обучения иностранному языку для специальных целей [4; 7].

Существенно, что эти стандарты создаются на английском языке и лишь частично описываются в переводной литературе. При нестандартных ситуациях разработчик вынужден обращаться к первоисточнику, и способность понимать формулировки официальных спецификаций – в частности, различать обязательные и рекомендательные требования (must, should, may) – становится частью профессиональной грамотности [6]. Английский язык сопровождает весь цикл разработки: на нём формулируются сообщения об ошибках, записи журналов, комментарии при рецензировании кода. Недостаточное владение языком замедляет поиск неполадок, а неверное понимание документации способно привести к ошибочным архитектурным решениям.

*Классификация англоязычных цифровых ресурсов.*

Многообразие доступных студенту англоязычных цифровых ресурсов целесообразно упорядочить по их функциональному назначению в учебном процессе. Такая классификация помогает преподавателю осознанно подбирать материалы под конкретные образовательные задачи, а студенту – понимать, к какому типу источника следует обращаться в той или иной ситуации. Основные группы ресурсов и их роль в подготовке разработчиков клиент-серверных систем представлены в таблице 1.

Таблица 1

Виды англоязычных цифровых ресурсов  
в подготовке разработчиков клиент-серверных систем

Тип ресурса	Примеры	Учебная функция
Официальная документация	Документация языков, фреймворков и библиотек, спецификации протоколов	Чтение аутентичных технических текстов, освоение терминологии
Образовательные платформы	Интерактивные курсы и руководства по веб-разработке	Структурированное изучение материала с языковой поддержкой
Профессиональные сообщества	Форумы вопросов и ответов, тематические обсуждения	Понимание неформальной профессиональной речи, поиск решений
Репозитории кода	Платформы совместной разработки, открытые проекты	Работа с описаниями проектов, комментариями и обсуждениями
Видео- и аудиоматериалы	Технические доклады, лекции, подкасты	Развитие навыков аудирования профессиональной речи

Представленная классификация носит рабочий характер: на практике границы между группами подвижны, и один и тот же ресурс может выполнять несколько функций. Тем не менее подобное упорядочивание позволяет выстроить последовательную траекторию работы – от чтения адаптированных учебных материалов к самостоятельному использованию оригинальной документации и участию в профессиональных обсуждениях.

*Языковые трудности при работе с англоязычными ресурсами.*

Как показывает педагогическая практика, освоение англоязычных профессиональных текстов студентами технологического вуза сопровождается рядом устойчивых трудностей. Эти трудности носят системный характер и не могут

быть преодолены простым увеличением объёма читаемых материалов: они требуют целенаправленной педагогической работы и должны учитываться при отборе ресурсов и проектировании заданий. Среди наиболее значимых трудностей выделяются лексико-семантические, грамматико-синтаксические и дискурсивные.

Лексико-семантические трудности обусловлены явлением межъязыковой паронимии («ложные друзья переводчика»). В сфере клиент-серверной разработки оно встречается особенно часто: *accurate* означает «точный», а не «аккуратный»; *commit* – «фиксация изменений», а не «обязательство»; *patch* – «исправление кода», а не «заплатка»; *bug* – «дефект программы», а не «жук». Отдельную группу составляют общеупотребительные слова, приобретающие специальное значение: *handler* – программный обработчик, *socket* – конечная точка сетевого соединения, *thread* – поток исполнения. Без целенаправленной работы с подобной лексикой студент рискует сформировать ошибочные представления о предметной области.

Грамматико-синтаксические трудности связаны с особенностями технического стиля. Официальные спецификации насыщены пассивными конструкциями (*the request is processed, the connection is established*), длинными цепочками определений (*server-side request validation logic*) и условными предложениями, описывающими реакцию системы. Особое значение имеет правильная интерпретация модальных глаголов: в спецификациях IETF *must* обозначает строго обязательное требование, *should* – настоятельную рекомендацию, *may* – допустимую возможность [6]. Дискурсивные трудности проявляются в работе с профессиональными сообществами: студенту необходимо различать жанры (официальный вопрос на форуме, неформальное обсуждение в чате, комментарий при рецензировании кода) и владеть соответствующими речевыми средствами.

*Методика использования ресурсов в обучении.*

Простое предоставление студентам доступа к англоязычным ресурсам не гарантирует формирования устойчивой профессиональной компетенции. Указан-

ные выше языковые особенности технических текстов нередко снижают мотивацию обучающихся, поэтому ключевой задачей преподавателя становится методическое сопровождение работы с такими материалами: отбор текстов по уровню сложности, предварительное снятие лексических трудностей, формулирование заданий, направленных на извлечение конкретной информации. Подобная организация учебного процесса опирается на общие положения теории обучения иностранным языкам, рассматривающей язык как средство деятельности, а не как изолированную систему правил [1; 2].

Целесообразно выстраивать работу по принципу постепенного усложнения: от чтения коротких фрагментов официальной документации и поиска в них ключевых терминов – к сравнению развёрнутых руководств из разных источников, а затем к продуктивной деятельности, то есть написанию собственных пояснений к коду на английском языке, формулированию вопросов в профессиональном сообществе и реферированию технических статей.

На основе изложенного можно предложить практический алгоритм работы с англоязычными цифровыми ресурсами на занятии. Сначала определяется профессионально значимая коммуникативная задача – например, разобраться в описании программного интерфейса, оформленном по спецификации OpenAPI [9], или сформулировать вопрос об ошибке. Затем подбирается соответствующий тип ресурса согласно таблице 1, проводится языковая работа (снятие терминологических трудностей и анализ типовых конструкций), после чего студент создаёт собственный речевой продукт – краткое изложение, комментарий или вопрос на английском языке. Завершает работу рефлексия, в ходе которой обучающийся объясняет, какие языковые средства он использовал и какие затруднения преодолел. На практике это реализуется через типовые профессиональные жанры: пересказ описания операции из справочника по программному интерфейсу, составление сообщения об ошибке для системы отслеживания задач, краткое изложение обсуждения из профессионального сообщества, написание понятных комментариев к фрагменту кода.

Особого внимания требует оценивание результатов такой работы. В качестве критериев целесообразно использовать точность передачи терминологии, корректность интерпретации исходного текста и ясность собственного высказывания на английском языке. Первичную проверку формальной правильности можно поручить автоматизированным средствам, тогда как содержательная оценка остаётся за преподавателем. Существенное преимущество такого обучения состоит в его двойном результате: студент одновременно осваивает предметное содержание – принципы построения клиент-серверного взаимодействия – и развивает иноязычную компетенцию в её профессионально значимом измерении. Косвенно востребованность такого подхода подтверждается отраслевыми опросами разработчиков, демонстрирующими широкое использование международных инструментов, языков программирования и платформ, освоение которых неотделимо от работы с англоязычными источниками [11].

#### *Заключение.*

Проведённое исследование позволяет сделать вывод, что англоязычные цифровые ресурсы являются необходимым компонентом подготовки разработчиков клиент-серверных систем. Английский язык выступает в этой сфере не вспомогательным, а рабочим инструментом, без владения которым невозможны доступ к актуальной профессиональной информации, работа с официальными стандартами и полноценное участие в процессе разработки. Предложенная классификация ресурсов – от официальной документации до профессиональных сообществ – даёт преподавателю ориентир для осознанного отбора материалов под конкретные учебные задачи.

Эффективность использования таких ресурсов определяется не их доступностью, а качеством методического сопровождения: учётом языковых особенностей технических текстов и официальных спецификаций, продуманным усложнением заданий, переходом от рецептивной к продуктивной деятельности и обоснованными критериями оценивания. Предложенный практический алгоритм позволяет связать работу с цифровыми ресурсами с реальной коммуникативной задачей. Интеграция профессионального содержания и иноязычной подготовки

даёт возможность формировать у студентов технологического вуза компетенцию, отвечающую реальным требованиям индустрии разработки программного обеспечения.

Научная новизна работы заключается в систематизации англоязычных цифровых ресурсов применительно к подготовке разработчиков клиент-серверных систем и в обосновании необходимости включения официальных международных спецификаций (RFC, OpenAPI, OWASP) в содержание иноязычной подготовки. Практическая значимость состоит в том, что предложенные классификация ресурсов, типология языковых трудностей и алгоритм работы на занятии могут быть непосредственно использованы преподавателями иностранного языка технологических вузов.

### *Список литературы*

1. Гальскова Н.Д. Теория обучения иностранным языкам: лингводидактика и методика: учеб. пособие / Н.Д. Гальскова, Н.И. Гез. – М.: Академия, 2017. – 368 с.
2. Образцов П.И. Профессионально-ориентированное обучение иностранному языку на неязыковых факультетах вузов: учеб. пособие / П.И. Образцов, О.Ю. Иванова. – Орёл: ОГУ, 2005. – 114 с. EDN SIGMAP
3. Соловова Е.Н. Методика обучения иностранным языкам: базовый курс: пособие для студентов педагогических вузов и учителей / Е.Н. Соловова. – М.: Просвещение, 2008. – 239 с. EDN QTOCEJ
4. Dudley-Evans T. Developments in English for Specific Purposes: A Multi-Disciplinary Approach / T. Dudley-Evans, M.J. St John. – Cambridge: Cambridge University Press, 1998. – 301 p.
5. Fielding R.T. Architectural Styles and the Design of Network-based Software Architectures: Doctoral dissertation / R.T. Fielding. – University of California, Irvine, 2000. – URL: <https://ics.uci.edu/~fielding/pubs/dissertation/top.htm> (date of access: 29.05.2026).

6. Fielding R. HTTP Semantics: RFC 9110 / R. Fielding, M. Nottingham, J. Reschke. – IETF, 2022. – URL: <https://www.rfc-editor.org/rfc/rfc9110> (date of access: 29.05.2026).
7. Hutchinson T. English for Specific Purposes: A Learning-Centred Approach / T. Hutchinson, A. Waters. – Cambridge: Cambridge University Press, 1987. – 183 p.
8. MDN Web Docs. Client-Server Overview. – Mozilla, 2025. – URL: [https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development/Extensions/Server-side/First\\_steps/Client-Server\\_overview](https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/First_steps/Client-Server_overview) (date of access: 29.05.2026).
9. OpenAPI Initiative. OpenAPI Specification. – 2024. – URL: <https://spec.openapis.org/oas/latest.html> (date of access: 29.05.2026).
10. OWASP. OWASP API Security Top 10. – 2023. – URL: <https://owasp.org/API-Security/> (date of access: 29.05.2026).
11. Stack Overflow. Developer Survey 2025. – 2025. – URL: <https://survey.stackoverflow.co/2025> (date of access: 29.05.2026).