

УДК 550.82

С.В. Корякин¹, srgkoryakin1@gmail.com

Д. Б. Орозова², orozovadariga@gmail.com

А. Т. Даутова², altynaidautova7@gmail.com

Б. Асакеева², bermetasakeeva96@gmail.com

¹Институт машиноведения автоматизации и геомеханики НАН КР

²Кыргызско-Германский институт прикладной информатики

МЕТОДЫ И ТЕХНОЛОГИИ РЕАЛИЗАЦИИ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ ПЛАТФОРМ ДЛЯ СЕМЕЙНОГО ОБРАЗОВАНИЯ В КЫРГЫЗСТАНЕ

Аннотация. В статье рассматриваются актуальные проблемы развития семейного образования в Кыргызской Республике и обосновывается необходимость создания специализированной цифровой образовательной платформы (ЦОС). Проведён комплексный сравнительный анализ пяти моделей разработки программного обеспечения (Waterfall, Agile, V-model, Spiral, Prototyping) и выявлены их принципиальные ограничения в контексте требований государственных образовательных стандартов (ГОС МОиН КР) и инфраструктурных условий регионов. Обоснована и формализована гибридная методология разработки, интегрирующая структурированный цикл Waterfall/V-model для регламентированного ядра системы, и методология Agile для адаптивной периферии. Предложена отказоустойчивая микросервисная архитектура с применением технологии Progressive Web Application (PWA), решающая критическую проблему нестабильного интернет-соединения в регионах. Описаны механизмы интеграции со СМЭВ «Түндүк» и локальными платёжными сервисами.

Ключевые слова: цифровая образовательная среда (ЦОС), модель жизненного цикла (МЖЦ), семейное образование, дистанционное обучение, микросервисная архитектура, PWA (Progressive Web App), проектирование информационных систем, СМЭВ «Түндүк», образовательные технологии (EdTech), Кыргызстан

Введение

В условиях цифровой трансформации общества происходят значительные изменения в системе образования. Традиционные формы обучения постепенно дополняются альтернативными моделями, среди которых особое место занимает семейное образование – организация учебного процесса вне традиционной школьной среды [17]. В последние годы интерес к данной форме обучения возрастает и в Кыргызстане, однако существующая образовательная инфраструктура ориентирована преимущественно на классическую школьную систему, что создаёт ряд трудностей для семей, выбирающих альтернативные форматы [1].

На практике родители вынуждены использовать разрозненные интернет-ресурсы и зарубежные образовательные сервисы – Moodle [5], Google Classroom [6], Khan Academy [7], которые не соответствуют государственным образовательным стандартам (ГОС МОиН КР) и не учитывают языковую специфику республики. Отсутствие единой цифровой среды значительно осложняет процесс организации семейного образования и снижает эффективность самостоятельного обучения.

Актуальной научной задачей является исследование методов и технологий реализации цифровых образовательных платформ, предназначенных для поддержки семейного образования в Кыргызстане. При этом ключевым является не только выбор технологического стека, но и обоснование методологии разработки, обеспечивающей одновременно соответствие государственным требованиям [3] и гибкость адаптации к реальным условиям пользователей.

Проблемы семейного образования в Кыргызстане

Несмотря на растущий интерес к семейному обучению, его развитие в Кыргызстане сталкивается с комплексом взаимосвязанных проблем инфраструктурного, методического и нормативного характера.

Центральной проблемой является **отсутствие специализированных цифровых образовательных платформ**, адаптированных к национальной системе образования [1]. Большинство доступных онлайн-ресурсов разработаны для международной аудитории и не учитывают требования ГОС МОиН КР [3], языковую специфику и нормы кыргызского законодательства об образовании.

Другой критической проблемой является **недостаток методической поддержки родителей**. В условиях семейного образования родители выступают организаторами учебного процесса, однако многие из них не располагают педагогическими инструментами для планирования обучения, контроля выполнения заданий и объективной оценки знаний учащихся.

Серьёзные технические ограничения связаны с **нестабильным интернет-соединением в ряде регионов страны**. В Нарынской, Баткенской и Ошской областях качество связи остаётся неудовлетворительным, что исключает возможность использования платформ, требующих постоянного подключения к сети [16].

Наконец, остро стоит проблема **юридической легитимности результатов семейного обучения**. Без официально признанной системы аттестации, интегрированной с государственными базами данных, результаты домашнего обучения не могут быть признаны в рамках формальной образовательной системы [1, 3].

Структура основных проблем семейного образования наглядно изображена на рисунке 1.



Рисунок 1 – Структура основных проблем семейного образования в Кыргызстане

В таблице 1 систематизированы проблемы, причины их возникновения и последствия для учебного процесса.

Таблица 1 – Основные проблемы семейного образования в Кыргызстане

№	Проблема	Причины возникновения	Последствия для обучения
1	Отсутствие специализированных ЦОС	Недостаточное развитие EdTech-сектора в КР	Сложность системной организации обучения
2	Недостаток методических ресурсов	Ограниченный объём адаптированного контента	Снижение качества учебного процесса
3	Нестабильный интернет в регионах	Неравномерное развитие цифровой инфраструктуры	Ограниченный доступ к онлайн-образованию
4	Отсутствие мониторинга успеваемости	Дефицит цифровых инструментов контроля	Невозможность объективной оценки результатов
5	Несоответствие иностранных платформ ГОС КР	Различия в стандартах и языковой среде	Юридическая нелегитимность результатов аттестации

Анализ существующих образовательных платформ

В настоящее время существует значительное количество цифровых образовательных платформ для организации дистанционного обучения. Среди международных решений выделяются Moodle [5], Google Classroom [6], Khan Academy [7] и Coursera.

Платформа **Moodle** представляет собой одну из наиболее распространённых LMS с открытым исходным кодом: позволяет создавать образовательные курсы, проводить тестирование и анализировать результаты [5]. **Google Classroom** ориентирована на интеграцию с облачными сервисами [6]. **Khan Academy** предоставляет видеоуроки, интерактивные упражнения и тесты [7].

Отдельно следует отметить национальную платформу «Тунгуч» [18], которая используется для поддержки дистанционного обучения и обеспечения доступа к образовательным ресурсам в рамках государственной цифровой инфраструктуры Кыргызской Республики. Она частично соответствует нормативным требованиям и обеспечивает базовую интеграцию с государственными сервисами, однако обладает ограниченной образовательной универсальностью и функциональной полнотой.

Несмотря на широкие функциональные возможности, сравнительный анализ (таблица 2) показывает, что ни одна из рассмотренных платформ в полной мере не отвечает совокупности требований, предъявляемых в условиях Кыргызской Республики: часть решений частично соответствует отдельным нормативным требованиям, включая государственные стандарты МОиН КР [3], однако ни одна не обеспечивает их комплексную реализацию, включая офлайн-работу и полную интеграцию с государственными системами [4]. Это обосновывает необходимость разработки специализированной национальной ЦОС.

Таблица 2 – Сравнительный анализ цифровых образовательных платформ

Платформа	Тип	Основные функции	ГОС КР	Офлайн	Нац. интеграция	Применимость
Moodle	LMS	Курсы, тест, контроль	Нет	Нет	Нет	Частично
Google Classroom	Образоват. платформа	Задания, материалы	Нет	Нет	Нет	Частично
Khan Academy	Онлайн-обучение	Видео, упражнения	Нет	Нет	Нет	Для контента
Coursera	Онлайн-курсы	Лекции, курсы вузов	Нет	Нет	Нет	Ограниченно
Тунгуч	Нац. платформа	Дистанционное обучение, доступ к материалам, инклюзивная поддержка	Частично	Частично	Да	Ограниченно (инклюзивная направленность)
Предлагаемая ЦОС	Нац. платформа	Полный цикл + аттестация	Да	Да (PWA)	Да (Тундук)	Полностью

Сравнительный анализ моделей разработки программного обеспечения

Характеристика основных моделей

Для обоснования выбора методологии разработки предлагаемой платформы был проведён анализ пяти ключевых моделей разработки программного обеспечения [12].

Модель **Waterfall** (каскадная) предполагает строгое последовательное выполнение этапов: сбор требований → проектирование → реализация → тестирование → внедрение. Главным преимуществом является прозрачность и документированность процесса. Ключевой недостаток – невозможность вносить изменения в требования после начала разработки без значительных затрат.

Модель **Agile** (Scrum/Kanban) основана на итеративной разработке короткими спринтами с непрерывной обратной связью [12]. Обеспечивает высокую адаптивность, однако затрудняет заблаговременное планирование бюджета и несёт риски при взаимодействии с жёстко регламентированными государственными требованиями.

Модель **V-model** является расширением Waterfall, в котором каждому этапу разработки соответствует этап тестирования. Обеспечивает высокое качество и раннее выявление дефектов, однако делает процесс ещё более жёстким в части изменений.

Спиральная модель (**Spiral**) структурирована как повторяющиеся циклы с анализом рисков на каждом витке [13]. Эффективна при высокой неопределённости требований, однако требует значительных ресурсов и управленческой зрелости команды.

Модель **Prototyping** предполагает быстрое создание прототипа для уточнения требований. Полезна на начальных этапах UX-исследований, однако без чёткого управления может привести к накоплению технического долга.

Полный сравнительный анализ представлен в таблице 3.

Таблица 3 – Сравнительный анализ методологий разработки для ЦОС КР

Модель	Принцип работы	Преимущества	Недостатки	Критический недостаток для ЦОС КР
Waterfall	Строгое последовательное выполнение этапов	Чёткость плана, полный контроль, документированность	Не допускает изменений после начала разработки	Обратная связь от регионов учтётся только через 12–18 месяцев
Agile (Scrum/Kanban)	Итеративные спринты с постоянной обратной связью	Гибкость, быстрое получение результата	Сложно планировать бюджет, риски при гос. регламентации	Аттестационный модуль может измениться в любом спринте – риск недействительности результатов
V-model	Тестирование планируется параллельно каждому этапу	Высокое качество, раннее выявление дефектов	Жёсткость, сложность изменений после начала	Адаптация к Нарынской области потребует полного перепроектирования
Spiral	Повторяющиеся циклы с анализом рисков	Управление рисками, гибкость	Высокая стоимость, требует экспертизы	Нереалистично для малых команд КР EdTech-рынка
Prototyping	Быстрое создание прототипа для проверки требований	Ранняя проверка концепции	Риск хаотичной разработки и технического долга	Критично для системы, хранящей данные несовершеннолетних [2]

Недостатки существующих подходов в контексте ЦОС КР

Аналитическое рассмотрение пяти моделей применительно к задаче разработки ЦОС для семейного образования в Кыргызстане позволяет выявить принципиальные ограничения каждого подхода.

- Waterfall не допускает итеративного уточнения требований, неизбежно возникающих в ходе апробации в регионах: обратная связь от родителей и учеников может быть учтена лишь через 12–18 месяцев.
- Agile в чистом виде допускает изменение функций аттестационного модуля в любом спринте, что создаёт риск нарушения нормативных требований МОиН КР и недействительности результатов аттестации [3].
- V-model полностью исключает адаптацию системы к специфике конкретного региона без полного перепроектирования.
- Spiral требует высокой управленческой зрелости и значительных ресурсов на оценку рисков, что нереалистично для малых команд разработки на рынке КР.
- Prototyping без надёжного управленческого каркаса ведёт к накоплению технического долга, что критично для системы, хранящей персональные данные несовершеннолетних [2].

Таким образом, ни одна из рассмотренных моделей в изолированном применении не способна обеспечить одновременно: соответствие ГОС МОиН КР, устойчивость при нестабильном интернете, адаптивность к региональным условиям и юридическую легитимность результатов аттестации. Это противоречие и стало основанием для разработки гибридной методологии.

Гибридный подход к разработке цифровой образовательной платформы

Концепция гибридного подхода

Под гибридной методологией разработки в настоящей работе понимается формализованная модель управления жизненным циклом программного обеспечения, в которой совокупность компонентов системы разбивается на два непересекающихся подмножества – регламентированное ядро и адаптивную периферию – с применением к каждому различных, но скоординированных методологических подходов.

Предложенный подход является технически обоснованным сочетанием методологий разработки, сформированным под воздействием двух противоречивых требований: необходимости строгого соответствия нормативным требованиям ГОС МОиН КР [3] и потребности в адаптивной настройке пользовательских функций для различных региональных условий.

Структура гибридного подхода: ядро и периферия

Гибридный подход структурно разделяет компоненты платформы на два уровня. Таблица 4 наглядно показывает, какой компонент разрабатывается по какой методологии и почему.

Таблица 4 – Распределение компонентов ЦОС по методологиям разработки

Компонент платформы	Методология	Причина выбора
Модуль аттестации (ГОС МОиН КР)	Waterfall + V-model	Изменения недопустимы без согласования с МОиН КР [3]
Интеграция со СМЭВ «Тундук»	Waterfall	Жёсткий протокол государственной системы [4]
Защита персональных данных	Waterfall + V-model	Требования Закона КР № 58 [2]
Хранение данных на серверах КР	Waterfall	Нормативное требование законодательства КР
Пользовательский интерфейс (UI/UX)	Agile/Scrum	Требует итеративной адаптации под регионы
Учебные задания и курсы	Agile/Scrum	Содержание меняется по обратной связи пользователей
Геймификация и рекомендации	Agile/Scrum	Функционал уточняется итеративно с каждым спринтом
Отчётность и аналитика	Agile/Scrum	Формат отчётов адаптируется к нуждам родителей
Уровень синхронизации	-	Связывает ядро и периферию, защищает от конфликтов

Для компонентов регламентированного ядра применяется следующая последовательность: составление полного технического задания → формальное проектирование → реализация → верификация по V-model → приёмочное тестирование с представителями МОиН КР → ввод в эксплуатацию. Изменение этих компонентов осуществляется только через процедуру управления изменениями с обязательной документацией.

Для адаптивной периферии работа организована в виде спринтов длительностью 1–2 недели. По завершении каждого спринта инкремент демонстрируется пользователям в пилотных регионах, собирается обратная связь и вносятся корректировки [12].

Схема гибридного подхода представлена на рисунке 2.

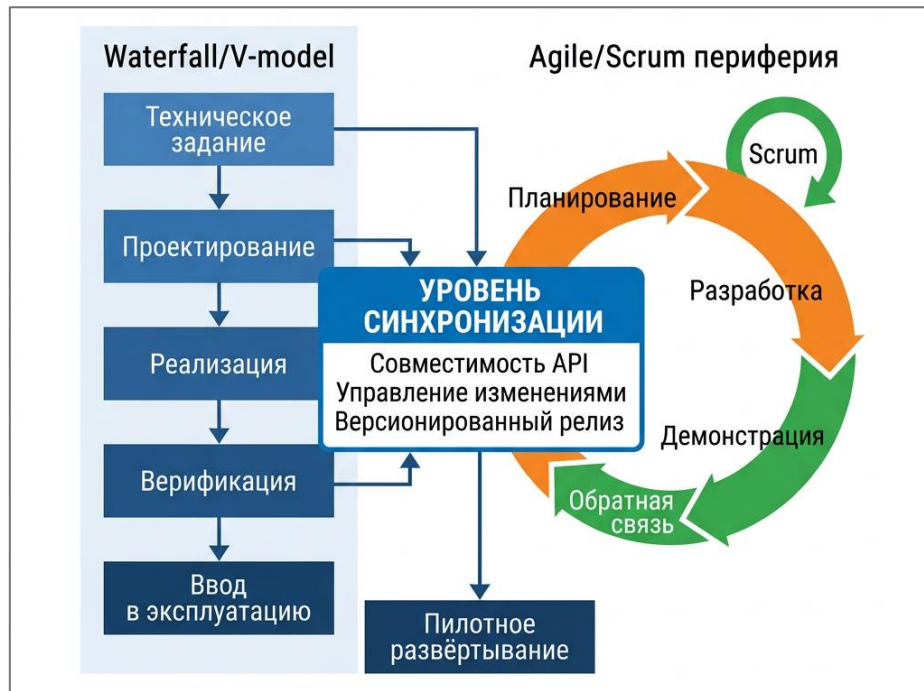


Рисунок 2 – Схема гибридного подхода к разработке ЦОС (два параллельных потока + уровень синхронизации)

На схеме левый поток – «Регламентированное ядро (Waterfall + V-model)» – отображает линейную последовательность: ТЗ → Проектирование → Реализация → Верификация → Валидация → Ввод в эксплуатацию. Правый поток – «Адаптивная периферия (Agile/Scrum)» – представлен циклом спринтов: Планирование → Разработка → Демонстрация → Обратная связь → (следующий спринт). Между двумя потоками расположен блок «Уровень синхронизации», в котором проверяется совместимость API, выполняется управление изменениями и формируется версионированный релиз. После синхронизации система переходит к пилотному развёртыванию, откуда обратная связь возвращается в поток Agile.

Механизм синхронизации

Ключевым элементом гибридного подхода является уровень синхронизации – процедура, обеспечивающая согласованность между ядром и периферией. Синхронизация проводится в конце каждого Agile-спринта по следующему протоколу:

1. Команда Agile формирует перечень изменений, затрагивающих API-интерфейсы компонентов ядра.
2. Технический архитектор проверяет совместимость изменений с зафиксированными требованиями к ядру.
3. При отсутствии конфликтов изменения интегрируются в общую кодовую базу.
4. При наличии конфликтов инициируется процедура управления изменениями для компонента ядра с обязательной документацией.
5. Результатом синхронизации является версионированный релиз, передаваемый на пилотное развёртывание.

Такой подход предотвращает ситуацию, при которой итеративные изменения в интерфейсе случайно нарушают работу аттестационных модулей или целостность данных.

Сравнение существующих подходов и предлагаемого решения

В таблице 5 представлено сравнение существующих подходов (отдельные методологии и зарубежные платформы) с предлагаемым гибридным решением по ключевым критериям. Это позволяет наглядно показать, в чём именно гибридная методология превосходит каждый из отдельно взятых подходов.

Таблица 5 – Сравнение существующих подходов и предлагаемого гибридного решения

Критерий	Существующие подходы	Предлагаемый гибридный подход
Соответствие ГОС КР	Иностранные платформы не адаптированы; аттестация нелегитимна	Ядро строго соответствует ГОС; интеграция со СМЭВ «Түндүк» обеспечивает официальное признание результатов [3, 4]
Работа без интернета	Все рассмотренные платформы требуют постоянного соединения	PWA + Service Workers: офлайн-кэширование материалов, синхронизация при восстановлении связи [8, 9]
Адаптация к изменениям	Waterfall/V-model не позволяют вносить изменения без срыва плана	Agile-часть обеспечивает итеративную адаптацию UI/UX в каждом спринте [12]
Поддержка кыргызского языка	Moodle, Google Classroom, Khan Academy – без нативной поддержки кыргызского алфавита	PostgreSQL с индексацией символов ө, ң, ү; мультязычный i18n-интерфейс [11]
Интеграция с нац. экосистемой	Отсутствует интеграция с МBank, О!Деньги, Элсом, Түндүк	RESTful API-интеграция с платёжными шлюзами КР и СМЭВ «Түндүк» [4, 15]
Масштабируемость	Монолитные архитектуры сложно масштабировать под регионы	Микросервисная архитектура позволяет независимо обновлять и масштабировать компоненты [13]

Методология и проектирование платформы

Проектирование с ориентацией на пользователя (UCD)

Проектирование логики платформы базируется на методологии проблемно-ориентированного проектирования (User-Centered Design, UCD) [14]. В архитектуре платформы семейного образования выделяются три ключевые роли:

- «Ученик»: акцент на вовлечение, геймификацию и интуитивно понятный интерфейс.
- «Родитель / Тьютор»: инструменты мониторинга успеваемости, аналитика и финансовый контроль.
- «Администратор / Аттестационная комиссия»: доступ к статистике и проверка соответствия учебного плана стандартам КР [3].

Структура взаимодействия ключевых ролей пользователей представлена на рисунке 3.

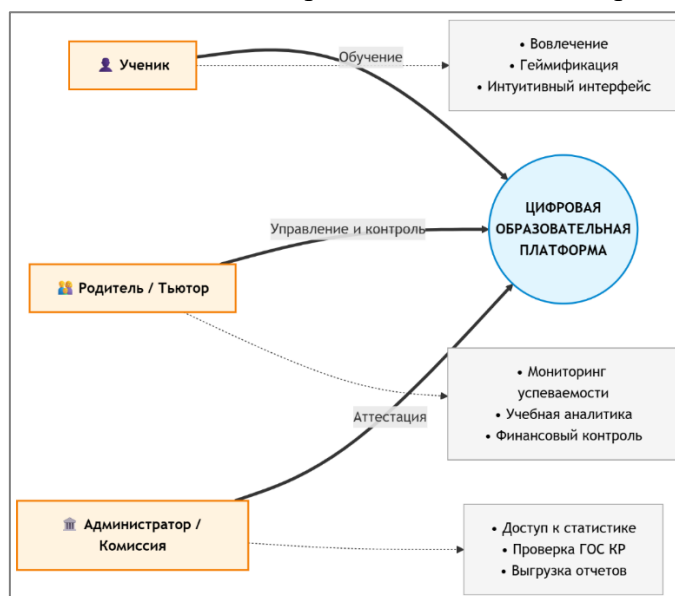


Рисунок 3 – Архитектура взаимодействия ролей пользователей в платформе семейного образования

Специфические требования к ЦОС в Кыргызстане

Помимо стандартных функциональных требований, к разрабатываемой системе предъявляется ряд специфических требований, обусловленных региональными особенностями:

- **Мультиязычность (i18n):** интерфейс и образовательный контент должны поддерживать кыргызский и русский языки в соответствии с законодательством КР [1].
- **Офлайн-доступность:** кэширование материалов для работы без интернета – критически важное требование для Нарынской, Баткенской и Ошской областей. Реализуется с помощью технологии Service Workers [9].
- **Информационная безопасность:** соблюдение Закона КР «Об информации персонального характера» № 58 от 14.04.2008 г. при хранении данных несовершеннолетних пользователей [2].

Технологии реализации платформы

Микросервисная архитектура

Для обеспечения высокой масштабируемости и отказоустойчивости системы предлагается использование микросервисной архитектуры [13]. Данный подход позволяет изолированно обновлять отдельные компоненты платформы (например, модуль тестирования) без остановки работы всей системы.

Схема микросервисной архитектуры цифровой образовательной платформы показана на рисунке 4.



Рисунок 4 – Микросервисная архитектура цифровой образовательной платформы

Предлагаемая архитектура включает следующие уровни:

Слой клиента (Client Layer) – пользовательские интерфейсы для учеников, родителей и администраторов. Платформа поддерживает работу через браузер, Progressive Web Application (PWA) [8] и мобильное приложение. PWA-технология на основе Service Workers обеспечивает частичный офлайн-режим: при пропадании связи ученик продолжает выполнять задания, результаты сохраняются локально и синхронизируются с сервером при восстановлении соединения [9].

API Gateway – единая точка входа для всех пользовательских запросов. Выполняет проверки доступа, маршрутизацию и распределение нагрузки между сервисами.

Слой микросервисов включает независимые сервисы: аутентификации пользователей, управления учебным контентом, проведения тестов, поддержки мультиязычного интерфейса и сбора аналитики. Поскольку сервисы работают независимо, их можно обновлять и масштабировать отдельно, не останавливая всю платформу [13].

Слой хранения данных (Database Layer) – база данных с профилями пользователей, курсами, заданиями и результатами тестов. В соответствии с законодательством КР серверы

размещаются на территории страны [2]; тяжёлые медиафайлы хранятся в S3-совместимом объектном хранилище.

Backend-технологии и базы данных

Серверная часть проектируется на базе языка Python с фреймворком FastAPI/Django [10]. Выбор Python обоснован широкими возможностями для интеграции алгоритмов машинного обучения, которые в перспективе будут формировать индивидуальные образовательные траектории.

Для хранения данных предлагается использовать СУБД PostgreSQL [11]. База данных оптимизируется и индексируется с учётом специфики кыргызского алфавита (ө, ң, ү) для обеспечения корректной работы системы полнотекстового поиска.

Интеграция с локальными сервисами

Для обеспечения функционирования платформы в экосистеме Кыргызстана архитектурой предусмотрена интеграция через RESTful API со следующими сервисами:

- **Платёжные шлюзы КР:** MBank, O!Деньги, MegaPay, Элсом – для удобной оплаты образовательных услуг.
- **WhatsApp Business API [15]:** автоматическая рассылка отчётов об успеваемости в формате push-уведомлений.
- **СМЭВ «Түндүк» [4]:** доверенная авторизация родителей через ЕСИ и официальное прикрепление учащихся к школам для прохождения государственных аттестаций.

Алгоритм проектирования цифровой образовательной платформы

На основе предложенной гибридной методологии разработан формализованный алгоритм проектирования ЦОС, определяющий последовательность и логику принятия решений на каждом шаге. Алгоритм структурирован в виде детерминированной процедуры с точками принятия решений (decision points), позволяющими направить процесс проектирования по ветви регламентированного ядра либо адаптивной периферии в зависимости от характеристик проектируемого компонента. Блок-схема алгоритма представлена на рисунке 5.

Алгоритм проектирования включает следующие шаги:

1. **Анализ предметной области.** Сбор и систематизация нормативных требований (ГОС МОиН КР [3], Закон № 58 [2], требования СМЭВ «Түндүк» [4]), изучение пользовательских сценариев для трёх ролей (ученик, родитель/тьютор, администратор), анализ инфраструктурных ограничений регионов.
2. **Классификация компонентов.** Для каждого проектируемого компонента выполняется проверка: подлежит ли компонент нормативному регулированию МОиН КР или законодательству КР? Если да – компонент относится к регламентированному ядру и проектируется по Waterfall/V-model. Если нет – к адаптивной периферии, проектируется в Agile-потоке.
3. **Параллельная разработка потоков.** Поток ядра выполняется линейно: составление ТЗ → формальное проектирование → реализация → верификация по V-model → приёмочное тестирование с МОиН КР. Поток периферии выполняется итеративно: планирование спринта → разработка → демонстрация пользователям пилотных регионов → сбор обратной связи → следующий спринт.
4. **Синхронизация потоков.** По завершении каждого Agile-спринта технический архитектор проверяет API-совместимость изменений с компонентами ядра. При обнаружении конфликта инициируется процедура управления изменениями. При отсутствии конфликта изменения интегрируются в общую кодовую базу и формируется версионированный релиз.
5. **Пилотное развёртывание и верификация.** Версионированный релиз развёртывается в пилотных регионах. Собранная обратная связь передаётся в Agile-поток для следующего цикла итераций либо при выявлении нормативных несоответствий, инициирует процедуру управления изменениями в потоке ядра.

Алгоритм проектирования цифровой образовательной платформы

для семейного образования в Кыргызской Республике

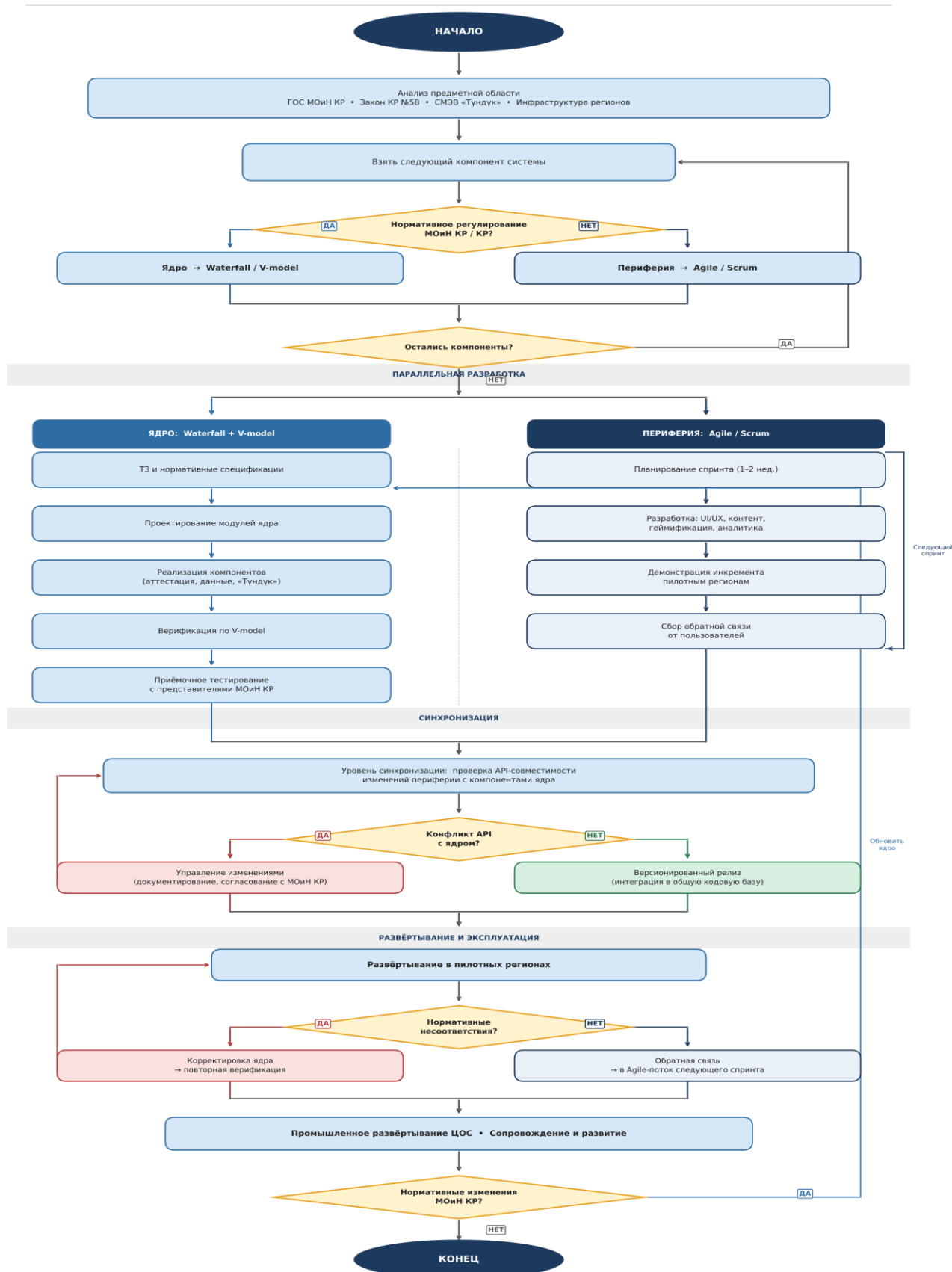


Рисунок 5 – Блок-схема алгоритма проектирования цифровой образовательной платформы

Этапы проектирования цифровой образовательной платформы

На основании разработанного алгоритма проектирование ЦОС для семейного образования структурировано в шесть последовательных этапов, каждый из которых имеет чётко определённые входные данные, выходные артефакты и критерии перехода к следующему этапу.

- 1. Этап предпроектного исследования.** Проводится комплексный анализ предметной области: изучение нормативно-правовой базы семейного образования в КР [1, 3], исследование потребностей целевых групп пользователей (родители, ученики, аттестационные органы), оценка существующих платформ (Moodle, Google Classroom, Khan Academy) и выявление их критических ограничений в контексте требований КР. Результатом этапа является аналитический отчёт с обоснованием необходимости разработки специализированной национальной ЦОС.
- 2. Этап выработки методологии.** На основе результатов предпроектного исследования обосновывается и формализуется гибридная методология разработки. Производится декомпозиция системы на регламентированное ядро и адаптивную периферию, определяется состав компонентов каждого подмножества, формализуется механизм синхронизации потоков. Результатом этапа является документ «Методологическое обоснование», закрепляющий принципы управления жизненным циклом системы.
- 3. Этап системного проектирования.** Разрабатывается концептуальная архитектура платформы: определяется состав микросервисов, проектируются API-интерфейсы между компонентами, формализуются требования к интеграции со СМЭВ «Түндүк» [4] и платёжными шлюзами КР, специфицируются требования к офлайн-функциональности на базе PWA и Service Workers [8, 9]. Формируется полное техническое задание (ТЗ) для компонентов регламентированного ядра. Результатом этапа является пакет проектной документации.
- 4. Этап детального проектирования.** Для компонентов ядра разрабатываются детальные спецификации модулей аттестации, защиты персональных данных и интеграции с государственными системами с обязательной верификацией на соответствие нормативным требованиям по V-model. Для компонентов периферии в рамках первых Agile-спринтов создаются UX-прототипы пользовательских интерфейсов, проводится пользовательское тестирование с представителями целевых групп из различных регионов.
- 5. Этап реализации и верификации.** Параллельно ведётся разработка компонентов ядра (Python/FastAPI [10], PostgreSQL [11]) и итеративная разработка компонентов периферии в двухнедельных спринтах. В конце каждого спринта выполняется синхронизация потоков согласно описанному алгоритму. Завершением этапа является прохождение приёмочного тестирования компонентов ядра с участием представителей МОиН КР.
- 6. Этап пилотного внедрения и сопровождения.** Платформа развёртывается на серверах, расположенных на территории КР в соответствии с законодательством [2]. Проводится пилотная эксплуатация в нескольких регионах с различным уровнем интернет-покрытия. Собирается количественная обратная связь об образовательных результатах. Данные пилота передаются в Agile-поток для непрерывного совершенствования адаптивной периферии системы.

Модель жизненного цикла цифровой образовательной платформы

На основе описанных этапов и алгоритма проектирования формализована модель жизненного цикла (МЖЦ) ЦОС для семейного образования. МЖЦ представляет собой интегрированную двухпотокową модель, описывающую полный цикл существования системы – от первоначального замысла до вывода из эксплуатации – с учётом специфики гибридной методологии.

МЖЦ включает следующие фазы:

1. **Фаза инициации (Initiation).** Определение целей системы, формирование пула стейкхолдеров (МОиН КР, родительские сообщества, разработчики), предварительная оценка реализуемости. Результат: устав проекта и предварительный реестр требований.
2. **Фаза анализа и планирования (Analysis & Planning).** Детальный анализ предметной области, декомпозиция на ядро и периферию, разработка плана-графика двух параллельных потоков, формирование архитектурного видения системы. Результат: SRS (Software Requirements Specification) для ядра и Product Backlog для периферии.
3. **Фаза проектирования (Design).** Разработка архитектурных спецификаций микросервисов, проектирование схем баз данных, API-контрактов, интерфейсных прототипов. Для ядра – детальное техническое задание с согласованием МОиН КР; для периферии – UX/UI прототипирование в первых спринтах. Результат: архитектурная документация и верифицированное ТЗ.
4. **Фаза реализации (Implementation).** Параллельная разработка: поток ядра реализует нормативно-обязательные модули (аттестация, интеграция с «Түндүк», персональные данные) с поэтапной верификацией; поток периферии реализует UI/UX, контентные модули, геймификацию и аналитику в итеративных спринтах. Синхронизация выполняется после каждого спринта. Результат: готовые к развёртыванию версионированные релизы.
5. **Фаза тестирования и верификации (Testing & Verification).** Для ядра – приёмочное тестирование с участием МОиН КР, нагрузочное тестирование, аудит информационной безопасности согласно Закону КР № 58 [2]. Для периферии – пользовательское тестирование в регионах, А/В-тестирование UI-решений. Результат: подтверждённое соответствие нормативным требованиям и приемлемое пользовательское качество.
6. **Фаза эксплуатации и развития (Operation & Evolution).** Промышленная эксплуатация платформы на серверах КР, мониторинг производительности и доступности, непрерывное совершенствование компонентов периферии в Agile-режиме. Изменения ядра производятся только через формальную процедуру управления изменениями с участием МОиН КР. Обратная связь от пользователей всех регионов систематически собирается и анализируется.
7. **Фаза вывода из эксплуатации (Decommission).** При принятии решения о замене платформы обеспечивается миграция данных пользователей и образовательных результатов в новую систему с сохранением юридической легитимности аттестационных документов. Фаза реализуется по процедурам Waterfall с обязательным согласованием с МОиН КР.

Ключевой особенностью предложенной МЖЦ является рекуррентность: фаза эксплуатации не является завершающей точкой, а переходит в непрерывный цикл итеративного улучшения периферийных компонентов, тогда как ядро остаётся стабильным до возникновения нормативных изменений. Это обеспечивает долгосрочную устойчивость платформы при сохранении гибкости адаптации к пользовательским требованиям.

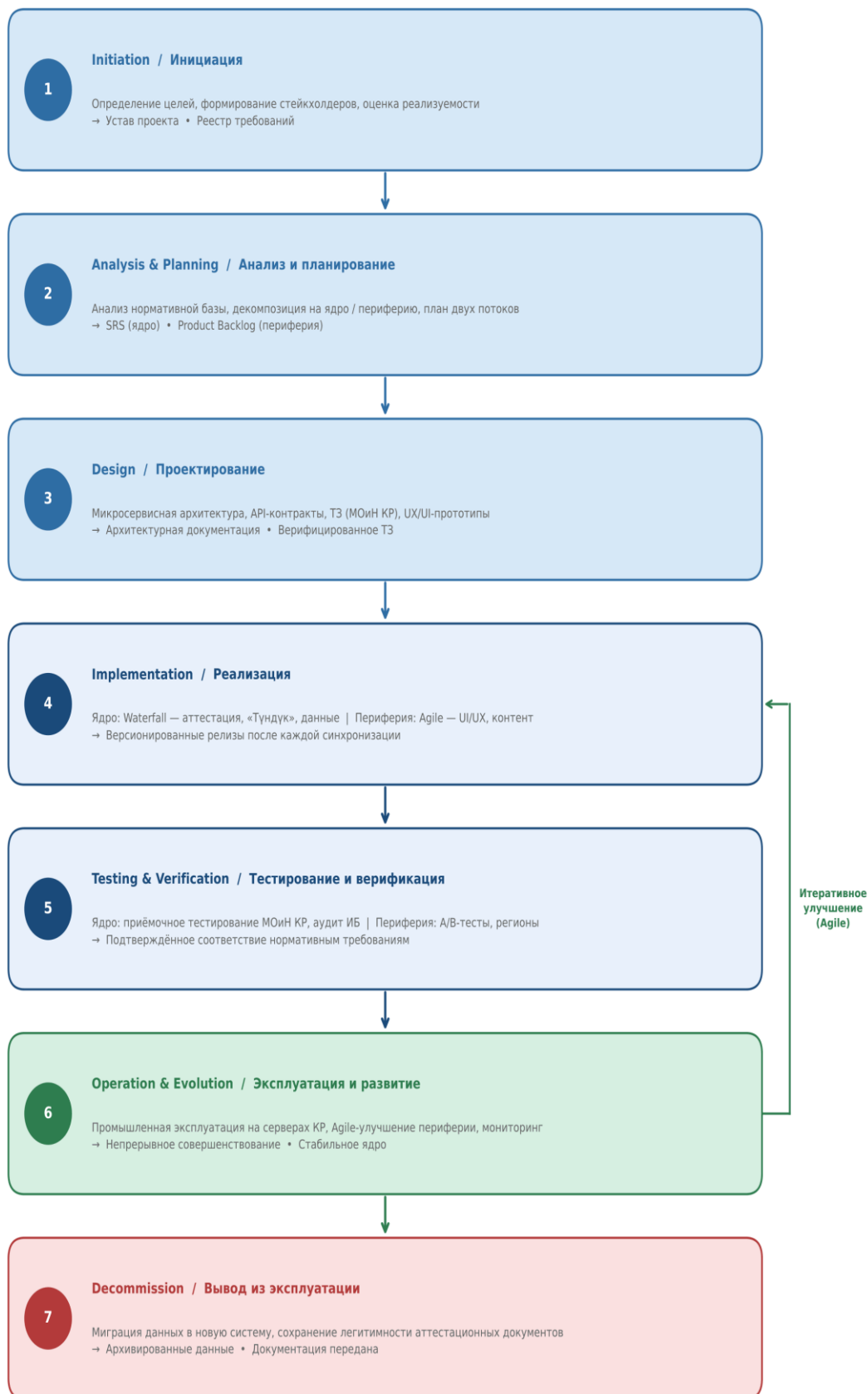


Рисунок 6 – Модель жизненного цикла цифровой образовательной платформы

Научная новизна

Научная новизна настоящего исследования состоит в следующем.

- 1. Предложена и формализована гибридная методология разработки ЦОС для Кыргызской Республики**, основанная на принципе разделения системы на регламентированное ядро (Waterfall/V-model) и адаптивную периферию (Agile/Scrum) с явным уровнем синхронизации. В отличие от работ, применявших данные методологии изолированно и без привязки к контексту государственного регулирования образования, предложенный подход обеспечивает одновременную выполнимость нормативных требований ГОС МОиН КР [3] и адаптивность к пользовательским условиям регионов.
- 2. Разработана архитектурная модель ЦОС семейного образования** с механизмом межкомпонентной синхронизации, интегрирующая микросервисный подход, PWA-технологии с автономным кэшированием (Service Workers) и RESTful API-взаимодействие со СМЭВ «Түндүк» и платежными сервисами. Предложенное решение обеспечивает непрерывность вычислительных процессов в условиях нестабильной связи и позволяет итеративно обновлять адаптивные пользовательские сервисы без нарушения целостности и юридической легитимности нормативно-аттестационного ядра платформы.
- 3. Проведён систематический сравнительный анализ пяти методологий разработки ПО** по критериям, специфичным для условий КР: соответствие ГОСТ, работа при нестабильном интернете, поддержка кыргызского алфавита, интеграция с национальными системами. Данный анализ может служить методической базой для проектирования образовательных ИС в схожих региональных контекстах.

Заключение

В ходе проведённого исследования решена актуальная научно-прикладная задача обоснования методологии проектирования и технологических принципов реализации цифровой образовательной платформы для поддержки семейного образования в Кыргызской Республике.

В отличие от существующих подходов, при которых либо нарушается соответствие ГОС (чистый Agile), либо исключается адаптация к регионам (чистый Waterfall), предложенная гибридная методология впервые обеспечивает выполнение обоих требований одновременно за счёт архитектурного разграничения регламентированного ядра и адаптивной периферии системы.

Установлено, что ни одна из рассмотренных международных платформ – Moodle [5], Google Classroom [6], Khan Academy [7] – не отвечает совокупности требований: ни одна не поддерживает ГОС МОиН КР [3], не обеспечивает офлайн-режим и не интегрирована с СМЭВ «Түндүк» [4]. Это подтверждает необходимость разработки специализированной национальной ЦОС.

Предложенная архитектурная модель, интегрирующая микросервисный подход [13], технологию PWA с офлайн-кэшированием [8, 9] и RESTful API-взаимодействие с государственными системами, обеспечивает практическую реализуемость методологии в условиях реального технологического ландшафта республики.

Практическая значимость результатов состоит в том, что разработанная модель и предложенная методология могут быть непосредственно использованы командами разработчиков при создании образовательных ИС, ориентированных на государственные стандарты КР и пользователей в регионах с ограниченной интернет-инфраструктурой [19,20].

Перспективы дальнейших исследований включают: разработку алгоритмов персонализированных образовательных траекторий на основе методов машинного обучения; пилотное внедрение платформы в нескольких регионах КР с количественной оценкой образовательных результатов; исследование применимости предложенной гибридной методологии к другим категориям государственных информационных систем республики.

Литература

1. Закон Кыргызской Республики «Об образовании». - URL: <https://cbd.minjust.gov.kg/act/view/ru-ru/202816> (дата обращения: 01.03.2025).
2. Закон Кыргызской Республики «Об информации персонального характера» № 58 от 14 апреля 2008 года. - URL: <https://cbd.minjust.gov.kg/act/view/ru-ru/202914> (дата обращения: 01.03.2025).
3. Государственный образовательный стандарт общего образования Кыргызской Республики. - URL: <https://edu.gov.kg/ru/normativno-pravovye-akty/> (дата обращения: 03.03.2025).
4. Система межведомственного электронного взаимодействия «Түндүк». - URL: <https://tunduk.gov.kg> (дата обращения: 05.03.2025).
5. Moodle - платформа дистанционного обучения. - URL: <https://moodle.org> (дата обращения: 05.03.2025).
6. Google Classroom. - URL: <https://edu.google.com/intl/ru/products/classroom/> (дата обращения: 06.03.2025).
7. Khan Academy. - URL: <https://ru.khanacademy.org> (дата обращения: 06.03.2025).
8. Google Developers. Progressive Web Apps. - URL: <https://web.dev/progressive-web-apps/> (дата обращения: 08.03.2025).
9. Mozilla Developer Network. Service Worker API. - URL: https://developer.mozilla.org/ru/docs/Web/API/Service_Worker_API (дата обращения: 08.03.2025).
10. FastAPI. - URL: <https://fastapi.tiangolo.com> (дата обращения: 10.03.2025).
11. PostgreSQL: документация. - URL: <https://www.postgresql.org/docs/> (дата обращения: 10.03.2025).
12. Beck K. et al. Manifesto for Agile Software Development. - URL: <https://agilemanifesto.org/iso/ru/manifesto.html> (дата обращения: 12.03.2025).
13. Newman S. Building Microservices: Designing Fine-Grained Systems. - O'Reilly Media, 2021. - 616 с.
14. User-Centered Design. - URL: <https://www.interaction-design.org/literature/topics/user-centered-design> (дата обращения: 14.03.2025).
15. WhatsApp Business API. - URL: <https://developers.facebook.com/docs/whatsapp> (дата обращения: 15.03.2025).
16. UNESCO ИТЕ. Цифровая трансформация образования. - URL: <https://iite.unesco.org/ru/> (дата обращения: 17.03.2025).
17. Семейное образование как форма получения начального общего образования. - URL: <https://cyberleninka.ru/article/n/semeynoe-obrazovanie-kak-forma-polucheniya-nachalnogo-obshchego-obrazovaniya> (дата обращения: 18.03.2025).
18. Онлайн-школа «Тунгуч» - URL: <https://tunguch.kg/>
19. Бактыбекова, Н. Б. Методы обнаружения и устранения проблем производительности для систем мониторинга и автоматической оптимизации запросов postgresql / Н. Б. Бактыбекова, С. В. Корякин, Э. А. Джалбиев // Проблемы автоматизации и управления. – 2025. – № 3(54). – С. 88–100. <https://www.elibrary.ru/item.asp?id=88914785>
20. Табылдаева А.А. Технологии и методы проектирования рекомендательных систем городских сервисов / А. А. Табылдиева, А. К. Кемелова, С. В. Корякин, Э. А. Джалбиев // Проблемы автоматизации и управления. – 2025. – № 3(54). – С. 71–87. <https://www.elibrary.ru/item.asp?id=88914784>