

ПРИМЕНЕНИЕ ЦИФРОВЫХ ТЕХНОЛОГИЙ В УПРАВЛЕНИИ ПРОЕКТАМИ: ИНТЕГРИРОВАННАЯ ВЕБ-ПЛАТФОРМА ПОДДЕРЖКИ AGILE-МЕТОДОЛОГИЙ НА ОСНОВЕ DJANGO И VUE.JS

*Хожамуратов Абат,
слушатель 2-го курса магистратуры
Высшей школы бизнеса и предпринимательства
при Кабинете Министров Республики Узбекистан*

Аннотация. В статье рассматриваются подходы к применению цифровых технологий в управлении проектами на примере проектирования интегрированной веб-платформы, поддерживающей Agile и Scrum. Акцент сделан не только на программной архитектуре, но и на управленческой ценности решения: повышении прозрачности проектного цикла, сокращении коммуникационных потерь, усилении контроля сроков и формировании единой цифровой среды принятия решений. В качестве технологической основы используются Django, Django REST Framework, Vue.js, PostgreSQL, Redis, MinIO, WebSocket и Docker. Приведены функциональные требования, архитектурные решения, меры информационной безопасности и результаты тестирования, демонстрирующие применимость платформы в условиях цифровой экономики.

Ключевые слова: цифровые технологии, управление проектами, Agile, Scrum, Django, Vue.js, контейнеризация, цифровая экономика, веб-платформа, проектная аналитика.

**BRINGING DIGITAL TO PROJECT MANAGEMENT: AN
INTEGRATED WEB PLATFORM FOR AGILE METHODOLOGIES
BASED ON DJANGO AND VUE.JS**

Abat Khozhamuratov,

**second-year master's student at the Higher School of Business and
Entrepreneurship under the Cabinet of Ministers of the Republic of
Uzbekistan**

Annotation. The article examines the application of digital technologies in project management through the design of an integrated web platform supporting Agile and Scrum methodologies. The focus is placed not only on software architecture, but also on the managerial value of the solution: transparency of the project cycle, reduction of communication losses, stronger deadline control and creation of a unified digital decision-making environment. Django, Django REST Framework, Vue.js, PostgreSQL, Redis, MinIO, WebSocket and Docker are used as the technological basis.

Keywords: digital technologies, project management, Agile, Scrum, Django, Vue.js, containerization, digital economy, web platform, project analytics.

Введение

Современные организации функционируют в условиях высокой динамики и неопределённости, что требует не просто электронных инструментов, а целостной цифровой среды для планирования, контроля и взаимодействия участников проектов. В дисциплине «Применение цифровых технологий в управлении проектами» такая среда рассматривается как управленческий механизм: она обеспечивает сбор данных о задачах, сроках, исполнителях, рисках и результатах, а также превращает эти данные в основу для принятия решений. Традиционные системы управления проектами (Jira, Trello, Asana) обладают развитой функциональностью, однако часто ограничены возможностями локальной интеграции, адаптации под национальные и корпоративные процессы, а также требованиями к размещению данных. В связи с этим актуальной становится разработка отечественных решений, ориентированных на Agile и Scrum.

1. Теоретико-методологические основы применения цифровых технологий в управлении проектами

Согласно PMBOK (PMI, 2021), управление проектом включает интеграцию процессов планирования, исполнения, мониторинга и завершения. ISO 21502:2020 дополнительно подчёркивает необходимость учитывать жизненный цикл проекта, заинтересованные стороны, коммуникации, ресурсы, риски и управление изменениями [3, 13]. Методологии Agile и Scrum, описанные Сазерлендом (2014) и Керцнером (2019), фокусируются на инкрементальной поставке ценности, самоорганизации команд и постоянной адаптации к изменяющимся требованиям. Поэтому цифровая платформа должна поддерживать не отдельную функцию, а полный управленческий цикл проекта.

2. Проектирование интегрированной веб-платформы поддержки Agile и Scrum

Проектирование веб-платформы осуществлялось на основе клиент-серверной архитектуры. Backend реализован на Django с использованием Django REST Framework, frontend — на Vue.js с применением компонентного подхода. В качестве базы данных выбрана PostgreSQL, Redis используется для кэширования, очередей уведомлений и обмена сообщениями, а MinIO — для распределённого хранения файлов. Развёртывание компонентов в Docker-контейнерах обеспечивает изоляцию сервисов, воспроизводимость окружения и более управляемый процесс CI/CD.

Основными принципами проектирования стали модульность, масштабируемость и отказоустойчивость. Коммуникация между сервисами реализована через REST и WebSocket-протоколы, что обеспечивает реактивное взаимодействие пользователей в реальном времени.

Также, согласно исследованию Халеевой Е.П. и др., веб-приложения подобного типа должны обеспечивать единое пространство для взаимодействия команд, хранения информации и коммуникации в режиме реального времени [2].

Основными функциональными требованиями для создаваемой системы определены:

- возможность создания и ведения проектов;
- организация ролей пользователей (автор, участник, эксперт, руководитель, инвестор);
- обмен сообщениями и уведомления в реальном времени;
- хранение файлов проекта в централизованной системе;
- работа приложения в изолированных контейнерах с минимальной зависимостью от окружения [6, 8].

Разработка выполнялась с использованием методологии Agile, что обеспечивает итерационный процесс поставки функционала. Основными принципами, реализованными в проекте, являются:

- инкрементальная разработка;
- тесное взаимодействие между разработчиками и пользователями;
- минимизация времени между релизами;
- прозрачность изменений и возможность быстрой адаптации к новым требованиям.

Методология Agile была выбрана в соответствии с рекомендациями К.В.Бугаковой, согласно которым гибкий подход позволяет обеспечить стабильность поставок и вовлечённость заказчика на всех этапах проектирования [1].

Каждая итерация включала анализ требований, разработку, тестирование, развёртывание и обратную связь. Средняя продолжительность спринта составляла 14 дней.

3. Архитектура и функциональные модули веб-платформы

Архитектура веб-приложения будет реализована по трёхуровневой модели:

- *Клиентская часть (Frontend)* — реализована на Vue.js. Используются компонентный подход и реактивная система обновления интерфейса.

Для визуализации применён фреймворк Vuetify, реализующий принципы Material Design.

- *Серверная часть (Backend)* — построена на Django с использованием Django REST Framework. Backend отвечает за аутентификацию, управление пользователями, взаимодействие с базой данных и REST API.
- *Хранилище данных* — основная база PostgreSQL, кэш Redis, файловое хранилище MinIO.
- *Коммуникация в реальном времени* — WebSocket, реализованный через Django Channels.
- *Инфраструктура* — все компоненты размещены в контейнерах Docker.

Подобная архитектура обеспечивает устойчивость к сбоям, простоту масштабирования и независимость отдельных сервисов.

Платформа включает несколько ключевых модулей: модуль пользователей и ролей, модуль проектов и задач, модуль обмена сообщениями и уведомлений, а также файловое хранилище. Аутентификация реализована через JWT-токены, что обеспечивает безопасность взаимодействия. Интерфейс, построенный на компонентном подходе Vue.js, обеспечивает интуитивную навигацию и визуализацию проектных процессов в формате Kanban-доски.

Основные компоненты интерфейса:

- *Главная панель* — навигация по проектам и уведомлениям;
- *Доска задач (Kanban)* — визуализация состояния задач;
- *Модуль сообщений* — чат, работающий через WebSocket;
- *Раздел хранения* — доступ к файлам через API MinIO;
- *Система ролей* — разграничение доступа для участников проектов.

Интерфейс реализует принципы UX-дизайна: адаптивность, интуитивность и унифицированный визуальный стиль.

4. Инфраструктурное обеспечение и цифровые сервисы платформы

Для управления структурированными данными используется PostgreSQL, обеспечивающая поддержку транзакций, индексов и запросов с высокой производительностью.

Redis применяется для хранения временных данных: токенов сессий, уведомлений и буфера сообщений. MinIO обеспечивает хранение файлов и изображений с интерфейсом, совместимым с Amazon S3.

Передача данных между компонентами организована через защищённые HTTP(S) и WebSocket-соединения.

Использование Docker позволило разделить сервисы на независимые контейнеры:

- **web** — Vue.js frontend;
- **api** — Django backend;
- **db** — PostgreSQL;
- **cache** — Redis;
- **storage** — MinIO.

Управление контейнерами осуществляется через **docker-compose**, что обеспечивает единое окружение для разработки и продакшена.

Для обеспечения CI/CD применена связка GitHub Actions и Docker Hub. При каждом изменении кода выполняется автоматическая сборка контейнеров и деплой на тестовый сервер.

Такой подход минимизирует риск несовместимости окружений и сокращает время развёртывания на 50–60 %.

Система безопасности реализована на уровне аутентификации JWT, шифрования данных в хранилище и разграничения прав пользователей.

Дополнительно обеспечены:

- резервное копирование PostgreSQL;
- дублирование Redis для отказоустойчивости;
- хранение файлов MinIO с избыточностью;
- мониторинг контейнеров с использованием Prometheus и Grafana.

Все соединения осуществляются по протоколу HTTPS с использованием TLS-сертификатов.

7. Результаты тестирования и оценка эффективности внедрения

Функциональное тестирование проводилось при помощи Postman и Pytest.

Результаты испытаний показали:

- среднее время отклика REST API — 110 – 150 мс;
- время инициализации WebSocket-соединения — до 200 мс;
- пропускная способность Redis — более 10 000 сообщений/с при нагрузке в 100 пользователей;
- корректная работа всех сервисов при горизонтальном масштабировании.

Нагрузочное тестирование подтвердило стабильность работы при одновременном подключении до 200 активных пользователей без потери производительности.

5. Цифровой контур управления проектом и его экономическое значение

С позиции цифровой экономики веб-платформа управления проектами должна рассматриваться не только как программный продукт, но и как инструмент снижения транзакционных издержек. В традиционной модели значительная часть времени руководителя и участников проекта расходуется на поиск актуальной информации, уточнение статусов, пересылку файлов, восстановление контекста после изменений и подготовку промежуточных отчётов. Интегрированная цифровая среда сокращает эти потери за счёт единого источника данных, автоматических уведомлений, журналирования действий и централизованного хранения документов.

Важным управленческим эффектом является повышение прозрачности проектного цикла. Kanban-доска показывает фактическое состояние задач, WebSocket-уведомления позволяют участникам оперативно реагировать на изменения, а история действий фиксирует, кто и когда изменил статус,

добавил комментарий или загрузил документ. Для руководителя проекта это означает переход от ручного контроля к управлению на основе данных. Для заказчика или инвестора это снижает информационную асимметрию и повышает доверие к результатам проектной команды.

Экономический результат внедрения платформы может оцениваться через систему показателей: среднее время постановки задачи, среднее время реакции на изменение статуса, долю задач, завершённых в срок, количество возвратов на доработку, время поиска проектного документа, интенсивность коммуникаций внутри проекта и уровень удовлетворённости пользователей. Эти метрики позволяют перейти от субъективной оценки удобства системы к измеримому анализу эффективности проектного управления.

Отдельного внимания заслуживает накопление проектных знаний. Комментарии, решения, статусы, файлы и показатели исполнения образуют цифровой след проекта, который может быть использован при планировании следующих инициатив. Организация получает возможность сравнивать плановые и фактические трудозатраты, выявлять типовые причины задержек, анализировать загрузку команды и повышать зрелость проектного управления. В перспективе на основе этих данных возможно внедрение аналитического модуля и элементов искусственного интеллекта для раннего выявления рисков просрочки.

6. Информационная безопасность и устойчивость платформы

Поскольку платформа хранит управленческие решения, документы, персональные данные участников и потенциально коммерчески значимую информацию, безопасность должна быть встроена в архитектуру с начального этапа проектирования. JWT-аутентификация обеспечивает защищённый доступ к API, ролевая модель ограничивает права пользователей, а HTTPS и WSS защищают каналы передачи данных. Для проектного управления это имеет принципиальное значение: потеря или искажение данных о задачах, сроках и решениях может привести не только к техническим сбоям, но и к управленческим ошибкам.

Контейнеризация повышает воспроизводимость инфраструктуры, однако требует дисциплины эксплуатации: регулярного обновления образов, разграничения сетей, хранения секретов вне исходного кода, резервного копирования PostgreSQL и файлового хранилища MinIO, а также мониторинга контейнеров. Использование Prometheus и Grafana позволяет контролировать состояние сервисов, нагрузку, ошибки и доступность компонентов. Таким образом, надёжность платформы становится частью общей системы управления рисками проекта.

Для условий Республики Узбекистан важны локализация интерфейса, возможность размещения платформы в национальной инфраструктуре и адаптация к требованиям организаций, реализующих проекты цифровой трансформации. Платформа может использоваться в образовательных учреждениях, государственных организациях и компаниях, где необходимо управлять распределёнными командами, обеспечивать прозрачность процессов и формировать культуру принятия решений на основе данных.

Заключение

В результате исследования разработана и апробирована веб-платформа, интегрирующая современные инструменты цифровизации и гибкие подходы к управлению проектами. Применение Django, Vue.js и Docker позволило создать масштабируемую и надёжную архитектуру, а реализация Agile-подхода обеспечила итерационное развитие и вовлечение пользователей. Полученные результаты демонстрируют, что внедрение подобных систем способствует повышению прозрачности, управляемости и экономической эффективности проектной деятельности. Для Республики Узбекистан такие решения имеют практическую значимость, поскольку поддерживают развитие отечественных цифровых продуктов, формируют культуру управления на основе данных и создают предпосылки для дальнейшей цифровой интеграции проектных процессов.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА:

1. Bugakova K.V. Development of a software design methodology using Agile technology. - Tolyatti: TSU, 2020. - 98 p.
2. Khaleeva E.P., Grebenkin A.R., Koptsev M.A. Development of a web service for project management // Research and Innovation. - 2023. - P. 96-100.3. PMI. A Guide to the Project Management Body of Knowledge (PMBOK Guide). — 7th Edition. Project Management Institute, 2021.
4. Kerzner, H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. — Wiley, 2019.
5. Sutherland, J. Scrum: The Art of Doing Twice the Work in Half the Time. — Crown Business, 2014.
6. Django Software Foundation. Django Documentation. — URL: <https://docs.djangoproject.com/>
7. Vue.js Core Team. Vue.js Guide. — URL: <https://vuejs.org/>
8. PostgreSQL Global Development Group. PostgreSQL Documentation. — URL: <https://www.postgresql.org/docs/>
9. Docker Inc. Docker Overview. — URL: <https://docs.docker.com/>
10. Redis Labs. Redis Documentation. — URL: <https://redis.io/documentation>
11. MinIO Inc. MinIO Architecture. — URL: <https://min.io/docs/>
12. Fowler M. Patterns of Enterprise Application Architecture. — Addison-Wesley, 2003.
13. ISO 21502:2020. Project, programme and portfolio management — Guidance on project management. — International Organization for Standardization, 2020.
14. PMI. The Future of Project Work: Pulse of the Profession 2024. — Project Management Institute, 2024.
15. Strategy “Uzbekistan - 2030”. Official portal of the Government of the Republic of Uzbekistan. — URL: https://gov.uz/en/pages/2030_strategy