

УДК 004.9

Фролова А.С.

студент

Ульяновский государственный технический университет

Россия, г. Ульяновск

Грачева Д.А.

студент

Ульяновский государственный технический университет

Россия, г. Ульяновск

Черкасова В.М.

студент

Ульяновский государственный технический университет

Россия, г. Ульяновск

ОБОСНОВАНИЕ АРХИТЕКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ СОПРОВОЖДЕНИЯ УНИВЕРСИТЕТСКИХ СТАРТАП- ПРОЕКТОВ

Аннотация: в статье рассматривается проектирование информационной системы сопровождения университетских стартап-проектов. Цель работы — выбрать архитектуру с учетом требований к доступности, масштабируемости и расширяемости. Используются IDEF0, UML и сравнительная оценка архитектурных решений. По результатам анализа клиент-серверная архитектура получила оценку 4,20 против 3,70 у монолитной и была реализована на базе 1С-Битрикс.

Ключевые слова: информационная система, проектирование, программная архитектура, стартап-проект, клиент-серверная архитектура, 1С-Битрикс..

Frolova A.S.

Student

Ulyanovsk State Technical University

Russia, Ulyanovsk

Gracheva D.A.

Student

Ulyanovsk State Technical University

Russia, Ulyanovsk

Cherkasova V.M.

Student

Ulyanovsk State Technical University

Russia, Ulyanovsk

JUSTIFICATION OF THE ARCHITECTURE OF AN INFORMATION SYSTEM FOR SUPPORTING UNIVERSITY STARTUP PROJECTS

Abstract: The article examines the design of an information system for supporting university startup projects. The aim of the study is to select an architecture considering system availability, scalability, and extensibility. IDEF0, UML, and comparative evaluation of architectural solutions were used. The client-server architecture received a score of 4.20 compared with 3.70 for the monolithic architecture and was implemented using 1C-Bitrix.

Keywords: information system, system design, software architecture, startup project, client-server architecture, 1C-Bitrix.

Введение

Развитие университетского технологического предпринимательства сопровождается накоплением значительного объёма информации о стартап-проектах. Для каждого из них необходимо хранить сведения о направлении работы, составе команды, текущем статусе, документах и результатах развития. Кроме того, требуется учитывать взаимодействие с партнёрами, инвестиции и гранты, а также формировать отчётность. По мере увеличения числа проектов работа с такими данными без специализированной информационной системы усложняется.

Предметная область исследования связана с сопровождением университетских стартап-проектов. Для каждого проекта в системе должна формироваться отдельная карточка с описанием, сведениями о команде и другой необходимой информацией. Отдельно учитывается партнёрская поддержка: данные о партнёрах, инвестициях и грантах связываются с конкретными проектами и используются при подготовке отчётности.

Для такой системы важна не только реализация требуемых функций, но и выбор архитектуры. Архитектурные решения принимаются на ранних этапах разработки и влияют на возможности дальнейшего изменения программного продукта, его производительность и сопровождаемость [1]. При этом универсальной архитектуры, одинаково подходящей для различных информационных систем, не существует. Выбор зависит от состава функций, предполагаемой нагрузки, требований к безопасности и дальнейшему развитию системы [2].

Система должна поддерживать разные категории пользователей и разграничение прав доступа: посетителям доступен просмотр стартапов, авторизованным пользователям — сопровождение проектов, а кураторам и администраторам — управление данными, партнёрской поддержкой и отчётностью. При этом архитектура должна обеспечивать дальнейшее расширение системы без полной переработки приложения.

Цель работы — обосновать выбор архитектуры информационной системы сопровождения университетских стартап-проектов и проверить принятое решение путем реализации программного прототипа.

Методы исследования

На первом этапе был проведён анализ предметной области и определены основные функции системы: ведение данных о стартап-проектах, участниках и партнёрах, учёт инвестиций и грантов, разграничение доступа, формирование отчётов и экспорт результатов.

Для описания процессов использовалось функциональное моделирование IDEF0, позволяющее представить функции системы, входные и выходные данные, управляющие воздействия и механизмы их выполнения [4]. На этой основе были определены связи между сопровождением проектов, партнёрской поддержкой и отчётностью.

Структура и поведение системы дополнительно описывались средствами UML. Использовались диаграммы вариантов использования, классов, последовательности, состояний, компонентов и развёртывания, отражающие структуру системы и взаимодействие её участников и компонентов [3].

Далее были сформированы требования к качественным характеристикам системы. При выборе архитектуры учитывались доступность, производительность, масштабируемость, расширяемость и сложность начальной реализации [1].

Для оценки архитектур были определены пять критериев. Наибольший вес получила доступность системы — 0,45, масштабируемость — 0,20, быстрый отклик и возможность добавления нового функционала — по 0,15, сложность первоначальной реализации — 0,05. Такое распределение обусловлено необходимостью постоянной доступности публичной части и дальнейшего развития системы.

В качестве альтернатив рассматривались монолитная и клиент-серверная архитектуры. Каждая из них оценивалась по пятибалльной шкале с учётом весов критериев, что позволило сопоставить решения в соответствии с требованиями конкретной системы.

Результаты оригинального авторского исследования

Результаты оценки показали, что по двум критериям архитектуры получили одинаковые показатели. Требование по доступности было оценено в 4 балла как для монолитного, так и для клиент-серверного

решения. Аналогичная оценка получена для времени отклика при типовой пользовательской нагрузке.

Основные различия проявились в масштабируемости и возможности развития функциональности. Монолитная архитектура получила 3 балла за масштабируемость, тогда как клиент-серверная — 5. Возможность добавления новых отчетов и сущностей была оценена соответственно в 3 и 4 балла. При этом монолитная архитектура оказалась проще на начальном этапе реализации и получила по данному критерию 5 баллов против 4 у клиент-серверной модели.

Критерий	Вес	Монолитная архитектура	Клиент-серверная архитектура
Доступность системы	0,45	4	4
Быстрый отклик	0,15	4	4
Масштабируемость	0,20	3	5
Добавление нового функционала	0,15	3	4
Сложность начальной реализации	0,05	5	4
Итоговая оценка	1,00	3,70	4,20

Таблица 1 — Сравнение архитектурных решений

Расчёт итоговых показателей составил 3,70 для монолитной и 4,20 для клиент-серверной архитектуры. Преимущество второго варианта связано прежде всего с более высокой масштабируемостью и расширяемостью, что важно при увеличении числа проектов и появлении новых видов отчётности.

Для дальнейшей реализации была выбрана клиент-серверная архитектура, в которой браузер отвечает за отображение информации и взаимодействие с пользователем, а обработка данных, проверка прав и бизнес-логика выполняются на сервере. Дополнительно применено слоистое разделение приложения, уменьшающее зависимость интерфейса от механизмов работы с данными и упрощающее сопровождение системы [1; 2].

Прототип реализован на платформе 1С-Битрикс. Серверная часть разработана на PHP с использованием Bitrix Framework, клиентская — на HTML, CSS и JavaScript. Для хранения данных используются информационные блоки и их свойства. Центральной сущностью является стартап-проект, связанный с участниками, ответственными лицами, документами, партнёрами, инвестициями и грантами.

В системе предусмотрено разграничение прав доступа. Публичная часть предназначена для просмотра проектов, а авторизованные пользователи могут редактировать данные, работать с партнёрами и формировать отчёты. Реализованы поиск, сортировка и фильтрация проектов, карточки стартапов с основными сведениями, документами и партнёрской информацией.

Отчётность включает формирование списка проектов на выбранную дату, анализ количества проектов по годам, сведения о партнёрской поддержке, грантах и инвестициях. Для финансовых показателей рассчитываются итоговые суммы, а результаты могут экспортироваться в офисные форматы.

В ходе реализации структура сущностей и связей была адаптирована к механизмам 1С-Битрикс, а формирование отчётов реализовано через серверные выборки. Разработанный прототип подтвердил возможность реализации основных сценариев и дальнейшего расширения системы без изменения общей схемы взаимодействия клиента и сервера.

Полученные оценки относятся к требованиям конкретной системы и не доказывают универсальное превосходство клиент-серверной архитектуры. Кроме того, в работе не проводилось полноценное нагрузочное тестирование, поэтому проверка требований к времени отклика и доступности требует отдельного исследования.

Заключение

В результате исследования была обоснована архитектура информационной системы для сопровождения университетских стартап-проектов. Основными требованиями к системе стали доступность, производительность, возможность масштабирования и дальнейшего расширения функциональности.

Для сравнения рассматривались монолитная и клиент-серверная архитектуры. С учетом установленных весов критериев итоговая оценка первого варианта составила 3,70, второго — 4,20. Преимущество клиент-серверной архитектуры связано прежде всего с более высокой оценкой масштабируемости и возможностью добавления новых функций без существенного изменения клиентской части.

Выбранное решение было реализовано в виде веб-приложения на базе 1С-Битрикс. Прототип обеспечивает ведение стартап-проектов и связанных с ними данных, учет партнеров, инвестиций и грантов, разграничение пользовательских прав и формирование отчетности. Реализация основных сценариев показала применимость выбранной архитектуры к поставленной задаче.

Практическая значимость работы состоит в возможности использования разработанного подхода при дальнейшем развитии университетской системы сопровождения стартапов. Расширение прототипа может быть связано с добавлением новых аналитических отчетов, интеграцией с другими университетскими информационными ресурсами и проведением нагрузочного тестирования для проверки количественных требований к производительности и доступности.

Использованные источники

1. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th ed. — Boston : Addison-Wesley Professional, 2021. — 464 p.
2. Крючкова Е. Н., Старолетов С. М. Объектно-ориентированное программирование: архитектурное проектирование и паттерны программирования : учебно-методическое пособие. — Барнаул : АлтГТУ, 2020. — 180 с.
3. Флегонтов А. В., Матюшичев И. Ю. Моделирование информационных систем. Unified Modeling Language : учебное пособие. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 112 с.
4. National Institute of Standards and Technology. Integration Definition for Function Modeling (IDEF0). Federal Information Processing Standards Publication 183. — Washington : U.S. Government Printing Office, 1993. — 95 p.