

THE ILLUSION OF FRONTEND: WHY SCALING IS MORE COMPLEX THAN IT APPEARS

Ubaydullayeva Dilorom Anorboevna

University of science and technologies, student of the specialty «software engineering»

E-mail: ubaydullayevadilorom22@gmail.com

Khasanov Abdushokhid Abdurashidovich

Director of the Registrar's Office at the University of Science and Technology,
Doctor of Philosophy (PhD) in Pedagogical Sciences, Associate Professor.

E-mail: abdushohid_1983@mail.ru

Abstract. This article examines the reasons why scaling frontend projects is a more complex task than it appears at the early stages of development. It analyzes architectural and organizational factors that influence the growth of complexity in user interfaces, including architectural choices, the expansion of the codebase, and team collaboration. Particular attention is given to monolithic and microfrontend approaches, as well as to the role of processes and tools in ensuring the sustainable evolution of frontend systems.

Keywords: fronted development, scalability, architecture, monolith, microfrontends, software engineering, technical debt, DevOps, team-based development.

ИЛЛЮЗИЯ ФРОНТЕНДА: ПОЧЕМУ МАСШТАБИРОВАНИЕ СЛОЖНЕЕ, ЧЕМ КАЖЕТСЯ

Убайдуллаева Дилором Анорбоевна [0009-0009-5013-9610]

Университет науки и технологий, студентка специальности
«программной инженерии»

Научный руководитель: **Хасанов Абдушоҳид Абдурашидович** [0000-0002-3940-0154]

Директор офиса Регистратора Университета науки и технологий,
доктор философии (PhD) по педагогическим наукам, доцент.

Аннотация. В статье рассматриваются причины, по которым масштабирование frontend-проектов представляет собой более сложную задачу, чем это кажется на начальных этапах разработки. Анализируются архитектурные и организационные факторы, влияющие на рост сложности пользовательского интерфейса, включая выбор архитектуры, увеличение кодовой базы и командное взаимодействие. Особое внимание уделяется монолитному и микрофронтенд-подходам, а также роли процессов и инструментов в обеспечении устойчивого развития frontend-систем.

Ключевые слова: frontend-разработка, масштабирование, архитектура, монолит, микрофронтенды, программная инженерия, технический долг, DevOps, командная разработка.

Введение. При старте веб-проекта фронтенд часто кажется простой и лёгкой частью, но постепенно выясняется, что это лишь «верхушка айсберга». Под поверхностью находятся системы сборки, маршрутизации, управления состоянием, требования к производительности и другие скрытые аспекты, без которых невозможно масштабировать приложение. При добавлении новых функций и с ростом команды разработчиков сложность проекта растёт, требуя продуманных архитектурных решений. Важно учитывать скрытые сложности с самого начала, чтобы обеспечить жизнеспособность. В статье рассмотрим, почему на поздних этапах разработки фронтенд-проект обрастает неожиданными трудностями и как этого избежать.

Архитектурный айсберг: простой интерфейс, сложные механизмы. Создать прототип фронтенд-приложения в современных условиях действительно не представляет сложности: развитые экосистемы React, Vue и Angular предоставляют готовые инструменты для быстрого запуска пользовательского интерфейса. Однако по мере расширения функциональности первоначально компактная кодовая база начинает усложняться. Увеличивается количество зависимостей, возрастает связность между компонентами, а процессы сборки и тестирования требуют всё больше ресурсов. Даже незначительные изменения могут затрагивать большое число модулей, что повышает риск ошибок и снижает скорость разработки.

На начальном этапе монолитная архитектура выглядит оправданной: она проста в реализации и не требует сложной инфраструктуры. Тем не менее при росте проекта монолит перестаёт масштабироваться эффективно. Разработчикам становится сложнее изолировать изменения, а добавление новых функций часто приводит к регрессиям в уже существующем функционале. Это особенно заметно при увеличении числа разработчиков, работающих с одной кодовой базой.

В качестве альтернативы всё чаще рассматривается подход микрофронтендов, при котором приложение разделяется на относительно автономные части. Такой подход облегчает параллельную разработку, снижает когнитивную нагрузку на команды и упрощает сопровождение отдельных модулей. Однако микрофронтенд-архитектура требует продуманной системы сборки, строгого управления версиями и согласованных процессов развертывания. Без этих условий архитектурная сложность может превысить ожидаемую пользу и снизить общую эффективность разработки.

Организационные сложности: команды, процессы, инструменты. По мере роста фронтенд-проекта координационные издержки становятся

критичными. Как отмечают аналитики McKinsey, когда несколько команд работают над одним кодом, без чёткого разграничения ответственности они не могут независимо вносить изменения возникают конфликты и перекрывающиеся задачи. Тесная связь компонентов «монолита» порождает технический долг, затрудняет интеграцию новых технологий и замедляет работу разработчиков. Единый конвейер сборки означает, что даже незначительные изменения требуют полной пересборки приложения, что увеличивает время цикла релизов и снижает гибкость команды. Например, в архитектуре микрофронтендов каждая команда отвечает за отдельную «вертикаль» — одну страницу приложения. Такое разделение позволяет команде выпускать обновления независимо, без ожидания синхронизации с другими командами. При этом необходимы гибкие DevOps-процессы: отдельные CI/CD-пайплайны, автоматизированные тесты и мониторинг качества. Без надёжных инструментов линтинга и тестирования масштабируемый проект быстро становится уязвимым.

Заключение. Итак, наращивание функционала фронтенда требует не только кода, но и продуманной архитектуры, организации работы и инструментов. С самого начала важно проектировать систему «под рост»: выбирать правильную компонентную структуру, разделять ответственность между модулями и командами, использовать автоматические проверки и развертывание. Важную роль играет мониторинг производительности и своевременное решение технического долга. Как подчеркивает Максим Вишнеvский, фронтенд — лишь «верхушка айсберга», а под капотом скрыты многочисленные системы и зависимости, с которыми приходится справляться при масштабировании. Современные практики разработки подтверждают, что пренебрежение архитектурой фронтенда ведёт к накоплению технической задолженности и замедлению разработки. Только сочетание технических

подходов (модули, микрофронтенды, оптимизация сборок) и гибких процессов (CI/CD, автоматическое тестирование, чёткое распределение задач) позволяет сохранить эффективность и контроль над проектом. Грамотное масштабирование фронтенда требует тщательного планирования: простота первых версий может обернуться значительными трудностями в будущем, если это не учесть заранее.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Чистяков В. Масштабирование фронтенд-приложений в 2023 году. NOP:Nuances of Programming (Medium), 2023.
2. Вишневецкий М. Просто о сложном: архитектура фронта для техлида. Habr, 2025.
3. Eizenman E., Villarreal O., Bhapkar R. и др. Need micro frontend benefits at scale? Reimagine the operating model. McKinsey, 2026.
4. Fowler M. Micro Frontends. martinowler.com, 2019.
5. Сайлор-Миллер К. Hard Problems in Front-End Platforms. QCon San Francisco (InfoQ), 2020.
6. Khasanov, A. A. (2017). Methods and methods of forming economic education through interdisciplinary communication through information technology. Journal, (3), 38.
7. Sharipov, D., Abdukadirov, A., Khasanov, A., & Khafizov, O. (2020, November). Mathematical model for optimal siting of the industrial plants. In 2020 International Conference on Information Science and Communications Technologies (ICISCT) (pp. 1-3). IEEE
8. Abdurashidovich, X. A., & Nigmanovna, M. F. (2019). Access to electronic educational resources in the education system. European Journal of Research and

Reflection in Educational Sciences Vol, 7(12).

9. Hasanov, A. A. (2020). Peculiarities of preparing teachers for the development and use of e-learning resources. *Theoretical & Applied Science*, (9), 15-17.

10. Khasanov, A. A. (2018). Didactic Foundations of Interdisciplinary Connections at Subject Teaching. *Eastern European Scientific Journal*, (6).

11. Hasanov, A. A., & Gatiyatulina, R. M. (2017). Interdisciplinary Communication as a Didactic Condition of Increasing the Efficiency of Educational Process. *Eastern European Scientific Journal*

12. Ravshanovna, P. N., & Abdurashidovich, K. A. (2019). Role of innovation in school development. *European Journal of Research and Reflection in Educational Sciences Vol*, 7(12).

13. Хасанов А.А. Современная теория обучения на межпредметной основе // *Science and world*. – Volgograd, 2016. -№8 (36), vol II. – С. 76-78

14. Khasanov A.A., Khasanova S.S. Theoretical approaches to the creation of pedagogical concepts // *American Journal of Pedagogical and Educational Research* 10 (2023): 185-190.

15. Khasanova, S. S. D., & Khasanov, A. A. (2023). Theoretical approaches to creation of pedagogical concepts. *Innovative Development in Educational Activities*, 2(6), 16-22.

16. Abdurashidovich, K. A. (2023). Methodological foundations of understanding the essence of e-learning. theory and analytical aspects of recent research, 2(13), 90-96.

17. Nazarov, I., Hasanov, A., Mirjamolova, F., Khaldarov, H., & Alibekov, S. (2021). Modern educational technologies. *Revista geintec-gestao inovacao e tecnologias*, 11(3), 245-252.