

Про дизайн, но технический. Зачем и как?



Щербаков Алексей

7 лет в коммерческой разработке

5 команд

7 продуктов

4 роли

тестировщик

разработчик

тимлид

наставник



🙏 [telegramm](#) | [linkedin](#) | [GitHub](#)

План

- Зачем нужен технический дизайн?
- Что такое технический дизайн и чем отличается от технической документации?
- В чём мой "чемпионский подход к ведению технических дизайнов"?
 - Как это работает?
 - Какие инструменты использованы?
- Итоги

Зачем нужен технический дизайн?

Контекст проекта

техника

процессы

Контекст. Техника

- компонент продукта
 - frontend
 - backend
 - QG
- дифференцированный стек
 - backend - Java
 - frontend - React
 - testing - Java, Selenium
 - devOps - Ansible, Jenkins

Контекст. Процессы

- разработка фичи "под ключ"
 - планирование
 - аналитика
 - реализация
 - поддержка
- инженеры на проекте могут меняться

Риски

- потеря связи с реальностью
- рассинхронизация
- дорогое вхождение в проект



Похоже на ваш проект?

Разрабатываете на React?

Что такое технический дизайн и чем отличается от технической документации?

This project was bootstrapped with [Create React App](#).

Available Scripts

In the project directory, you can run:

npm start

Runs the app in the development mode.

Open <http://localhost:3000> to view it in the browser.

The page will reload if you make edits.

You will also see any lint errors in the console.

npm test

Launches the test runner in the interactive watch mode.

See the section about [running tests](#) for more information.

npm run build



Builds the app for production to the `build` folder.

Readme.md - Техническая документация

как разрабатывать

как пользоваться

RFC React. Формат

Summary

...

Basic example

...

Motivation

...

Detailed design

...

Drawbacks

...

Alternatives

...

Adoption strategy

...

How we teach this

...

Unresolved questions



RFC. Процесс

- создание
- обсуждение
- финальное решение
- активен
- реализован

интернет стандартизируют через RFC

- RFC1 был опубликован 1969 году
- семейство протоколов TCP/IP описано в RFC

Технические дизайн. Маппинг на процесс

- планирование - концепция
- аналитика - уточнение концепции, детальный анализ
- реализация - корректировки
- поддержка - дизайн может быть использован в документации

Технический дизайн и техническая документация снижают риски

риск	решение
потеря связи с реальностью	дизайн-ревью
рассинхрон	дизайн-ревью
онбординг	чтение дизайнов и документации

Технический дизайн вещь хорошая. Будем использовать!



уже не аутсайдер, но ещё не чемпион

- ⌚ фиксация, структура, ревью на чек-боксах в confluence
- ⌚ 20+ дизайнов
- ⌚ обеспечение процесса ревью

чемпион

- 😊 дублирование ушло за счёт переиспользования
- 😊 автоматизация ведения технических дизайнов
- 😊 разработка технического дизайна как разработка кода



? Вопрос ?

Есть ли у вас друг подруги друга, который всё пытается представить как код в Git?

поднимите руку , если Да

Что угодно как код

- стандартизуемость
- масштабируемость
- переносимость
- аудит

Demo aka 🍖Мяско © Татьяна / 🍊Мякотка © Инна

Разработка через технический дизайн коней в вакууме

Результаты

- легкодоступная фактура для сверки с первоначальной задачей
- синхронизация в рамках обычного процесса разработки
- никаких "исторически так сложилось"

Инструменты технического дизайна

Диаграммы - сердце дизайна

лучше 1 раз увидеть, чем 100 раз прочитать

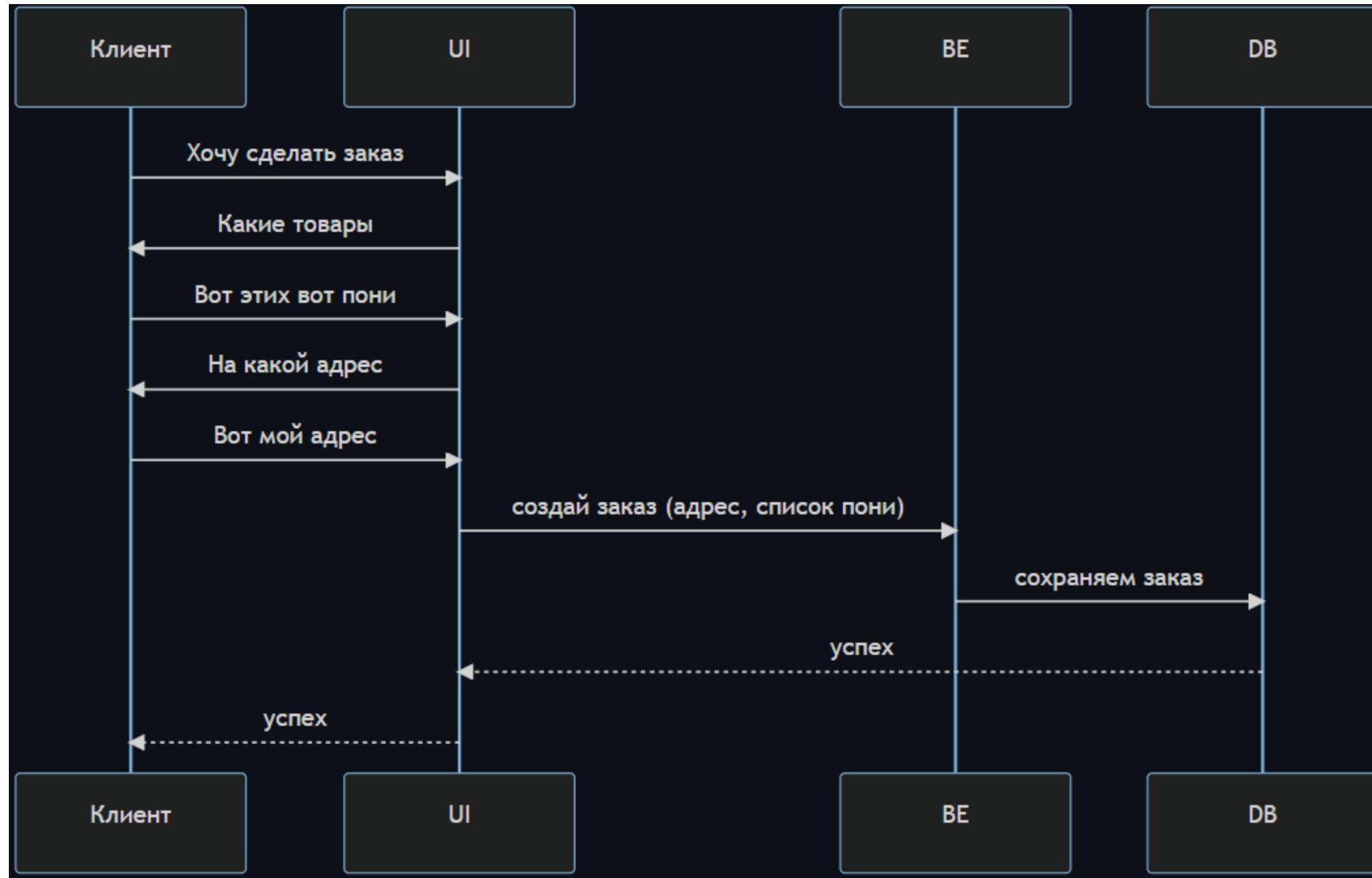
диаграммы - Mermaid

1. Написано на JS ❤️
2. [расширяемость, динамика](#) 👍
3. Поддерживает ZenUML 😎

Динамические аспекты поведения системы

Диаграмма последовательностей

- сценарии от пользователя
- сценарии программного взаимодействия



Архитектура

- диаграмма компонент

PlantUML

```
@startuml  
[UI] <--> Пользователь  
[Backend] - https  
[DB] - jdbc  
[Backend] <--> jdbc  
[UI] <-> https  
@enduml
```

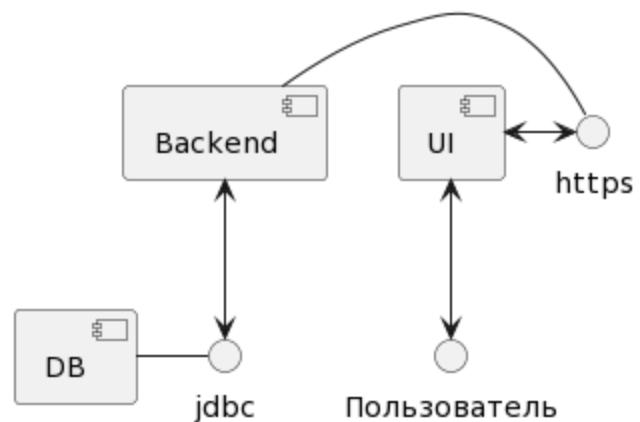


Схема базы данных

Диаграммы «сущность-связь»

mermaid - ERD

erDiagram

```
CUSTOMER ||--o{ ORDER : places
```

```
CUSTOMER {  
    int customerId  
    string name  
    string phone
```

```
}
```

```
ORDER {  
    int orderId  
    string deliveryAddress
```

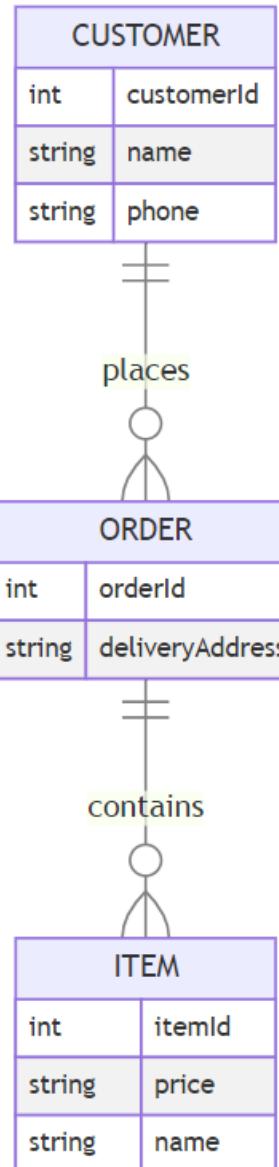
```
}
```

```
ORDER ||--o{ ITEM : contains
```

```
ITEM {  
    int itemId  
    string price  
    string name
```

```
}
```

```
%% add comments after that
```



Описание API

Openapi

```
description: FAIL, приложени  
/orders:  
  get:  
    Try it | Audit  
    operationId: orders-get-all  
    tags:  
      - ORDER  
    summary: получить список заказов  
    responses:  
      "200":  
        $ref: "#/components/respo  
components:
```

200 Список заказов

Media type **application/json** ▾
Controls Accept header.

Example Value | [Schema](#)

```
▼ [Order ▾ {  
  description: Заказ  
  orderId  
  orderNumber  
  deliveryAddress  
}]
```

OrderId number
example: 10005
▶ [...]
▶ [...]

Текст

Asciidoc

```
1 = Hello, AsciiDoc!
2
3 This is an interactive editor.
4 Use it to try https://asciidoc.org[AsciiDoc].
5
6 == Section Title
7
8 . A list item
9 * Another list item
10 . Another Another list item
11
12 [,ruby]
13 -----
14 puts 'Hello, World!'
15 -----
```

Hello, AsciiDoc!

This is an interactive editor. Use it to try [AsciiDoc](https://asciidoc.org).

Section Title

1. A list item
 - Another list item
2. Another Another list item

```
puts 'Hello, World!'
```

Презентации

- бизнес 😊 презентации
- презентация лучше скроллинга
- инструменты Marpit / Revealjs / \$hyoo_slides

MarplIt

```
---  
marp: true  
---  
  
# Первый слайд  
Контент первого  
![Картишка второго слайда](https://placehold.co/600x400/EEE/31343C)  
![bg right:50%](https://placehold.co/600x400/EEF/31343C)  
  
---  
# Второй слайд
```

Первый слайд

Контент первого

600×400

600×400

Автоматизация - plop

- Шаблонизация - handlebars
- CLI - Inquirer.js

ХУКИ В СЕРВЕРНОМ КОДЕ

Piterjs 59

Холивар: Глобальный Стейт || Локальные Сигналы

я покажу серверный код на хуках в одном из своих докладов



○ ○ ○

Inquirer.js/input/index.mjs

```
const [status, setStatus] = useState<string>('pending');
```

Почему такие инструменты использованы?

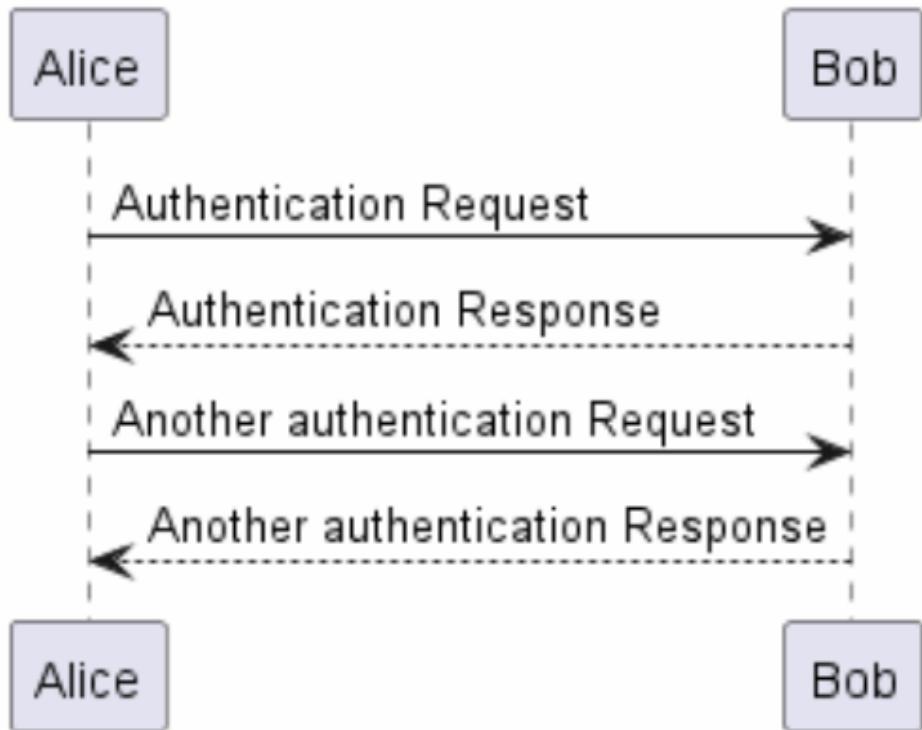
- идём от базовых потребностей
- должно быть не хуже чем в wiki
- избегаем вендорлока
- избегаем экзотических решений тегов сердцу ближе

◆ Диаграммы - PlantUml

1. стандарт дэ facto !
2. [plantuml.js](#) (WASM) 😞

◆ Диаграммы - PlantUml

```
@startuml  
Alice -> Bob: Authentication Request  
Bob --> Alice: Authentication Response  
  
Alice -> Bob: Another authentication Request  
Alice <-- Bob: Another authentication Response  
@enduml
```



Итоги

Напутствие

- делайте технические дизайны
- подстраивайте формат и тулинг под свои нужды

Никогда не делайте так 😊

Перемещено из другого репозитория

 Open Гендальф requested to merge into develop 1 minute ago

Commits 1 Changes 999+

 initial commit 

Без них не было бы этого доклада 😊

кому	за что
орг. Комитет Piterjs, в особенности Илье Государеву	прогоны, возможность выступить
Nexign Academy , в особенности Татьяне Кульбякиной	прогоны
 жене, дочке и сыну	

Ссылки

- [akaguny/rfc-of-a-horse-in-a-vacuum](#) - гитхаб репозиторий "абстрактного коня в вакууме"
- [RFC driven development](#)

