

АРХИТЕКТУРА. ДРУГОЙ ВЗГЛЯД



Жизнь надо прожить так, чтобы не было мучительно
больно за бесцельно прожитые годы

Я должен писать код так, чтобы не пришлось
переписывать ВСЁ, если резко поменяются требования

Хоршая архитектура должна быть гибкой!

SOLID, DRY, YAGNI, KISS

SOLID, DRY, YAGNI, KISS

Don't repeat yourself


```
<div className="ib-checkbox">  
  <input type="checkbox" className="ib-checkbox--input">  
</div>
```

```
function bemClass(block: string) {
  return (element?: string, mods?: object): string => {
    if (!element) { return block; }
    const result = [getElementClass(block, element)];
    if (mods) {
      forEach(mods, (needAdd, mod) => {
        if (!needAdd) { return; }
        result.push(getElementClass(block, element, mo
      ));
    }
    return result.join(' ');
  };
}
```

```
const bem = bemClass('checkbox');
```

```
export default () => (  
  <div class={bem()}>  
    <input type="checkbox" className={bem('input')}>  
  </div>  
)
```



```
const bem = bemClass('checkbox');

classnames({
  [bem('input')]: true,
  [bem('input', ['checked'])]: this.props.checked
})
```

```
`ib_checkbox--input  
${checked ? 'ib_checkbox--input__checked': ''}`
```

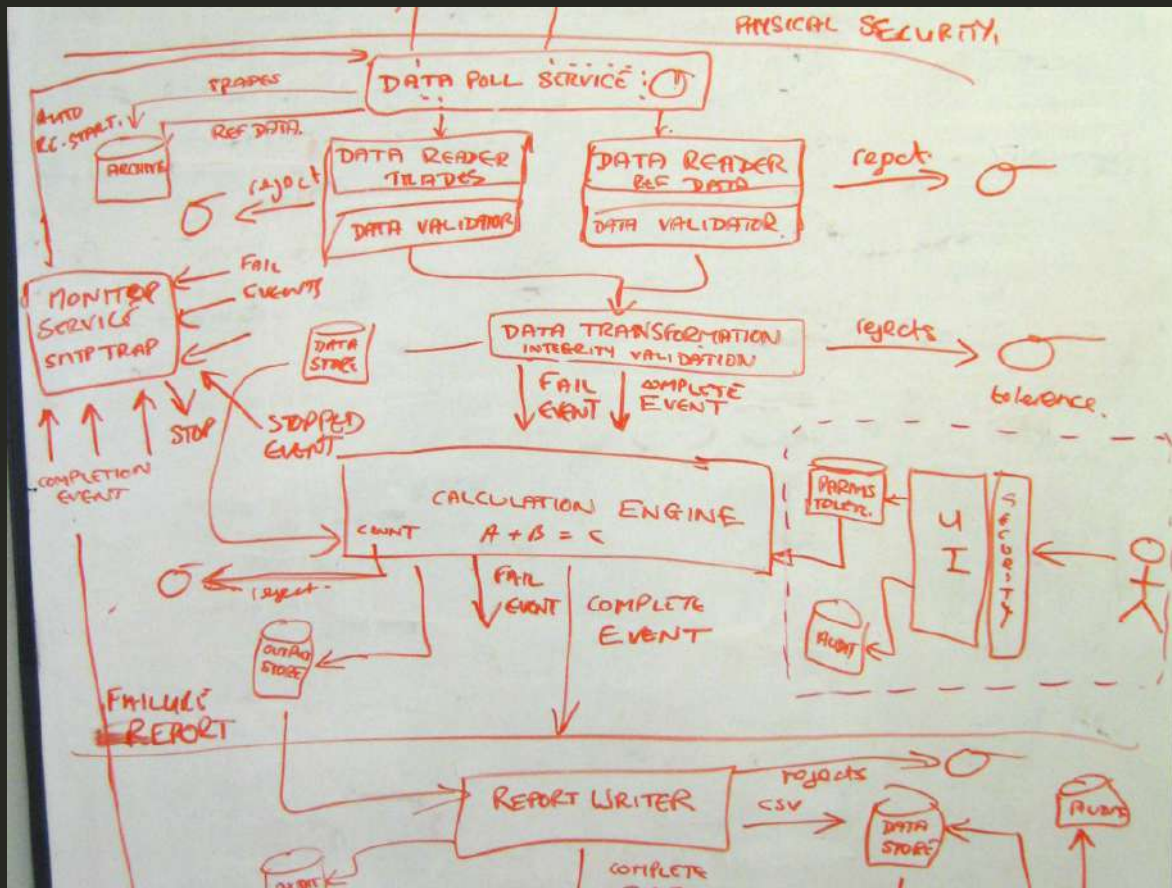

Есть DRY

Есть DRY, но еще есть KISS

Гибкость VS Сложность

Самый гибкий фреймворк – язык программирования

Причем здесь архитектура, ты же про код говоришь?!



Архитектура в IT — метафора

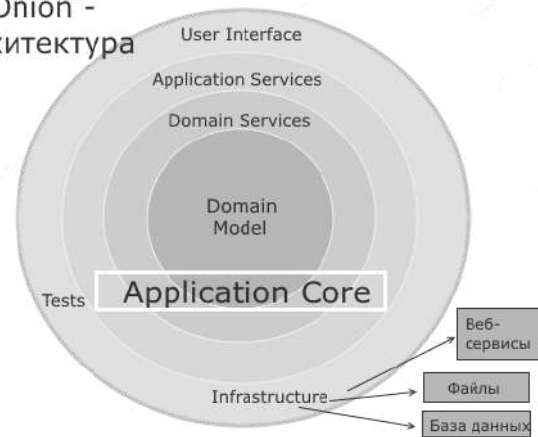
Архитектура в IT — метафора

Метафора, которой не было...

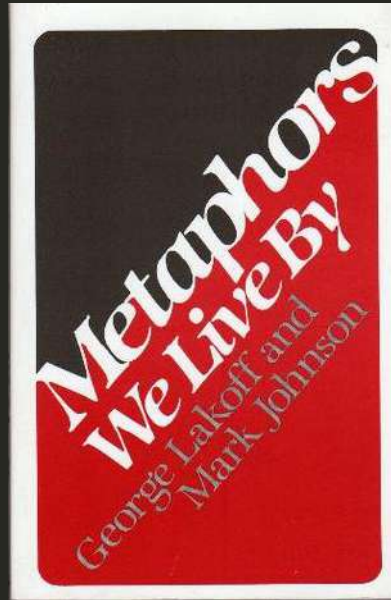


Луковая архитектура

Onion - Архитектура



METAPHORS WE LIVE BY



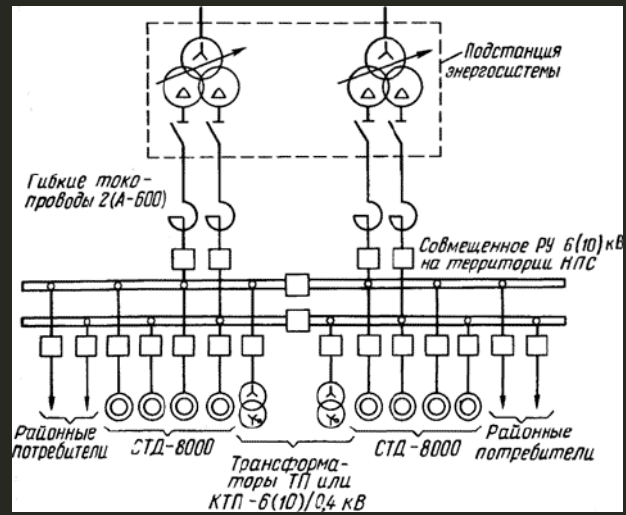
George Lakoff and Mark Johnson

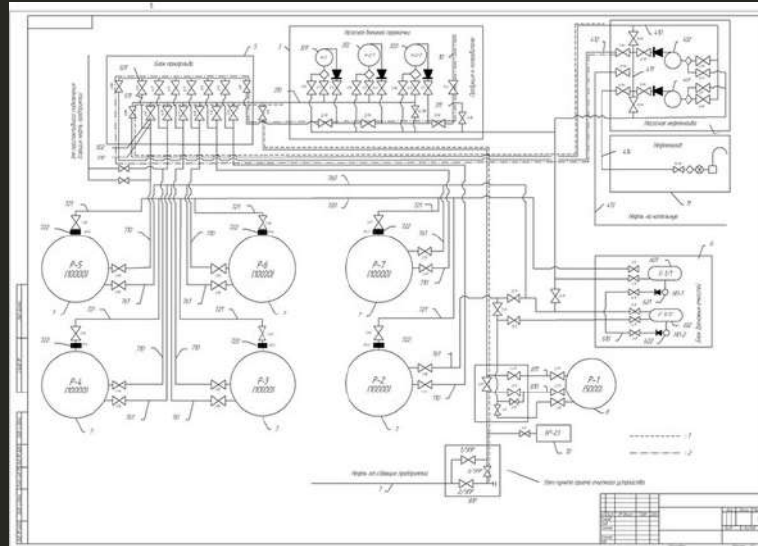


Давайте спланируем перед тем как делать!

Архитектура = План = Дизайн

Как все устроено в проектировании





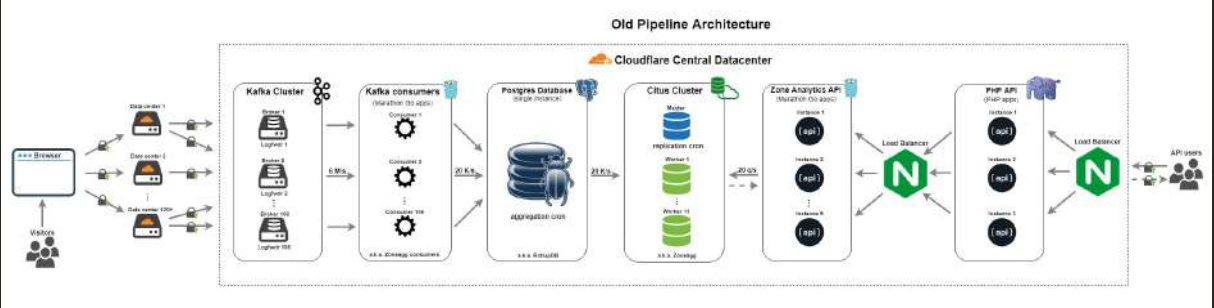




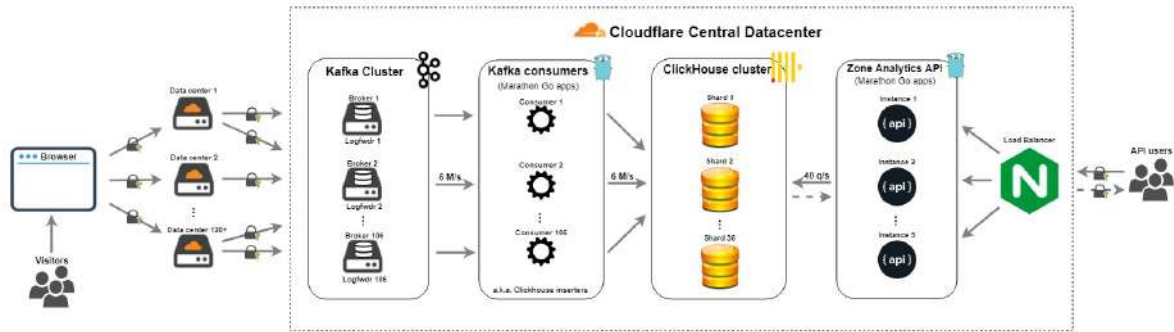
Вернемся в IT

Архитектура — совокупность важнейших решений об организации программной системы


[https://ru.wikipedia.org/wiki/
Архитектура_программного_обеспечения](https://ru.wikipedia.org/wiki/Архитектура_программного_обеспечения)















New Pipeline Architecture



kostyakulakov.ru



[Add site](#) [Support](#) [kostya1998-1998@yandex.ru](#)



[Overview](#) [Analytics](#) [DNS](#) [Crypto](#) [Firewall](#) [Speed](#) [Caching](#) [Page Rules](#) [Traffic](#) [ScrapeShield](#) [Customize](#) [Apps](#)

Overview

kostyakulakov.ru

Status: Active

This website is active on CloudFlare.


Quick Actions


Under Attack Mode


Development Mode

Advanced

Settings Summary

 **Security Level:** Medium
SSL: Flexible

 **Caching Level:** Standard
Development Mode: Disabled

 **DNS Records:** 7 Records
On CloudFlare: 6 Records

Subscription

Plan: Free Website

Change Plan

[What We Do](#)

[Plans](#)

[Overview](#)

[Community](#)

[Blog](#)

[Case studies](#)

[Support](#)

[Help Center](#)

[System Status](#)

[About Us](#)

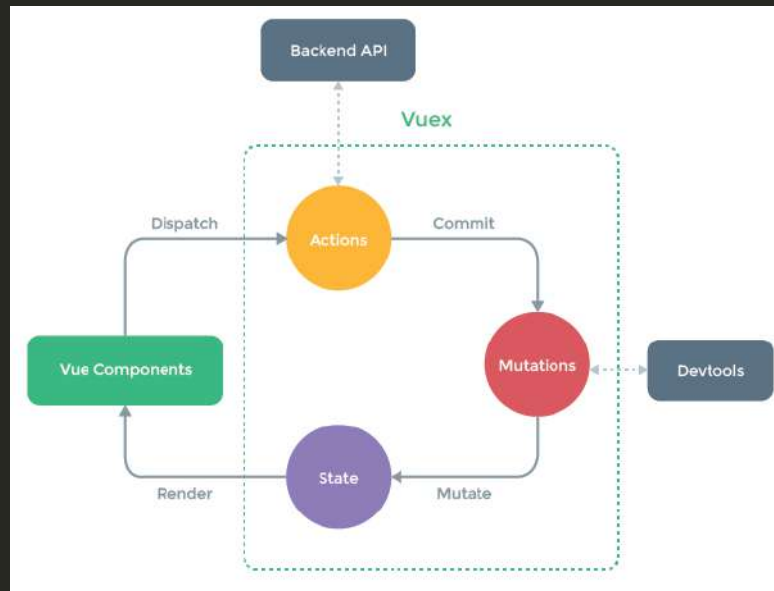
[Our team](#)

[Careers](#)

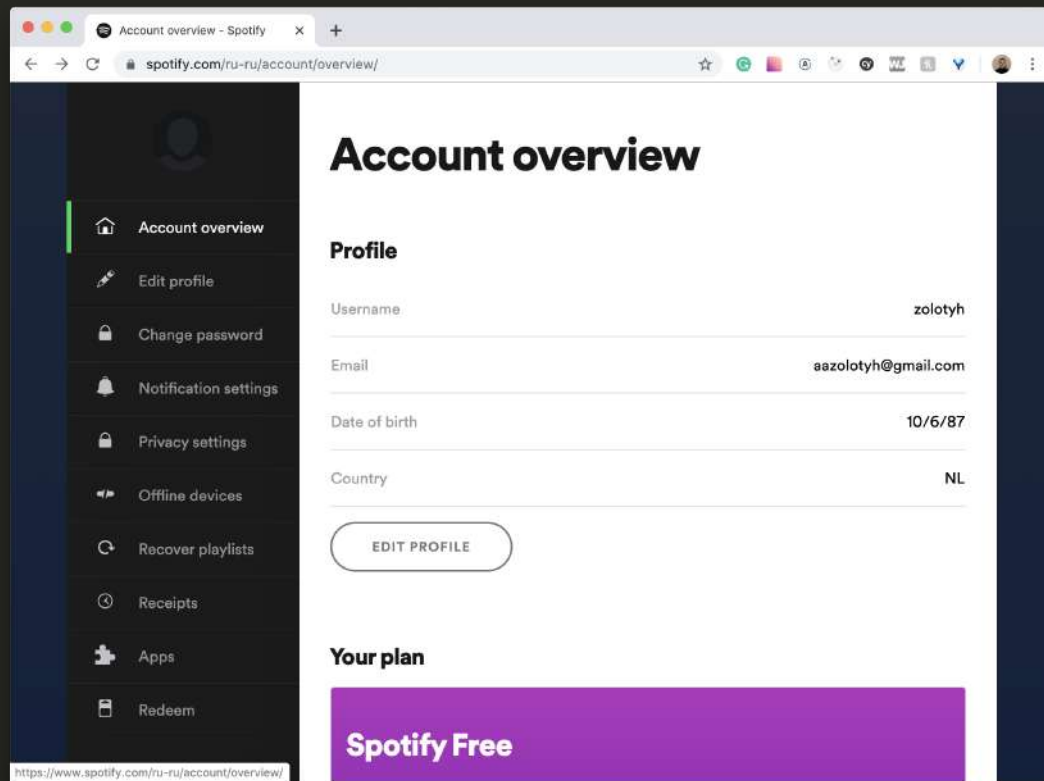
[Contact](#)

[Contact support](#)

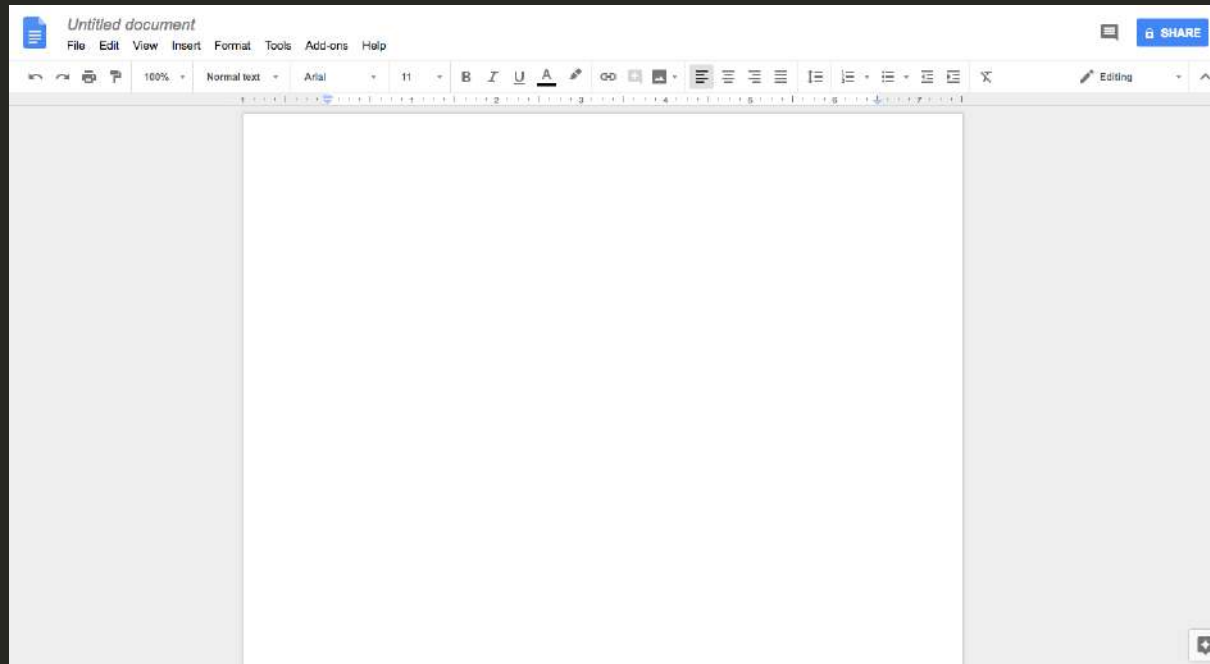
[Contact sales](#)



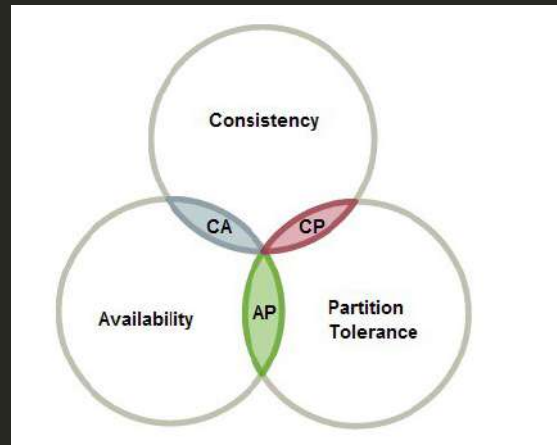
Vuex



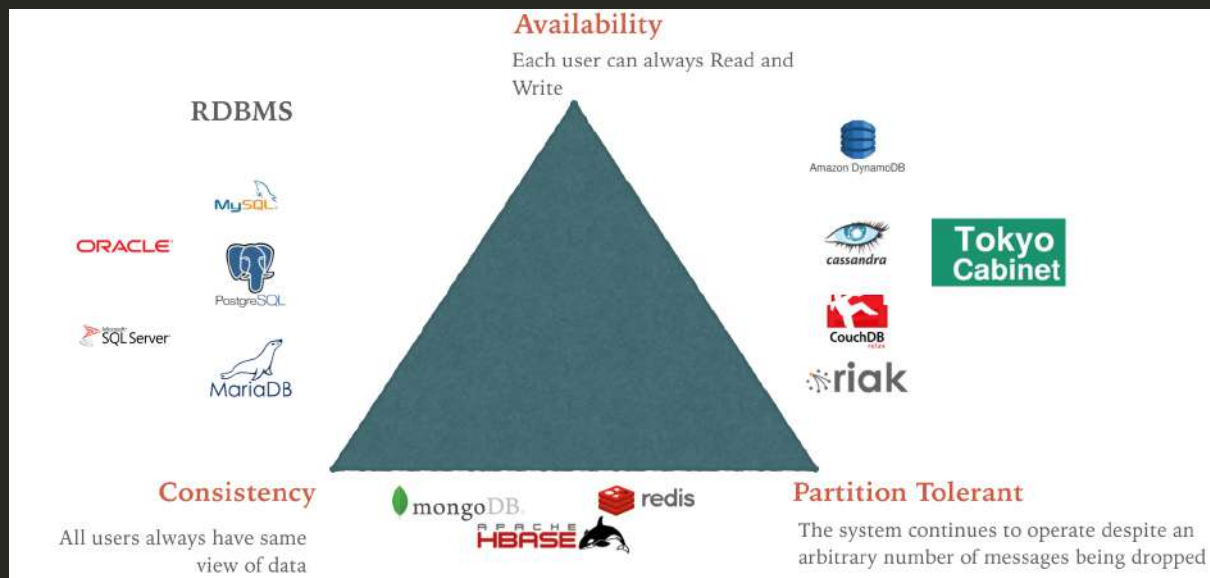
Важные – это те, которые невозможно поменять



Теорема CAP (известная также как теорема Брюера)



Теорема CAP (известная также как теорема Брюера)



Eventual consistency

en.wikipedia.org/wiki/Eventual_consistency

Operational transformation

ru.wikipedia.org/wiki/Операциональное_преобразование

en.wikipedia.org/wiki/Operational_transformation

Operational transformation определит архитектуру

Доклад про все это

Никита Прокопов — Клиенту и серверу нужно поговорить

youtu.be/uHGeI9_fgUw

Почему мы начали с алгоритма

ПРО ЛЕНЬ

ПРО ЛЕНЬ

Lean

Lean — концепция управления производственным предприятием, основанная на постоянном стремлении к устранению всех видов потерь

Имеет смысл откладывать ваши решения, если они не
критичны

Лишний дизайн = потери

Со временем информации становится больше

Robert C Martin – Clean Architecture and Design
(database is a detail)

youtu.be/Nsjsiz2A9mg?t=2541

От чего может зависеть архитектура
Евгений Кривошеев, Архитектура как функция от?

youtu.be/K-FtLWc8TxI

От чего может зависеть архитектура

От чего может зависеть архитектура

- Внешние требования

От чего может зависеть архитектура

- Внешние требования
- Внутренние требования

От чего может зависеть архитектура

- Внешние требования
- Внутренние требования
- Регламенты, стандарты

От чего может зависеть архитектура

- Внешние требования
- Внутренние требования
- Регламенты, стандарты
- Вопросы доверия и другие человеческие факторы

От чего может зависеть архитектура

- Внешние требования
- Внутренние требования
- Регламенты, стандарты
- Вопросы доверия и другие человеческие факторы
- Структура команды

От чего может зависеть архитектура

- Внешние требования
- Внутренние требования
- Регламенты, стандарты
- Вопросы доверия и другие человеческие факторы
- Структура команды (Закон Конвея)

От чего может зависеть архитектура

- Внешние требования
- Внутренние требования
- Регламенты, стандарты
- Вопросы доверия и другие человеческие факторы
- Структура команды (Закон Конвея)
- Время, ресурсы

От чего может зависеть архитектура

- Внешние требования
- Внутренние требования
- Регламенты, стандарты
- Вопросы доверия и другие человеческие факторы
- Структура команды (Закон Конвея)
- Время, ресурсы
- Законы, корпоративные ограничения

Разные точки зрения на архитектуру

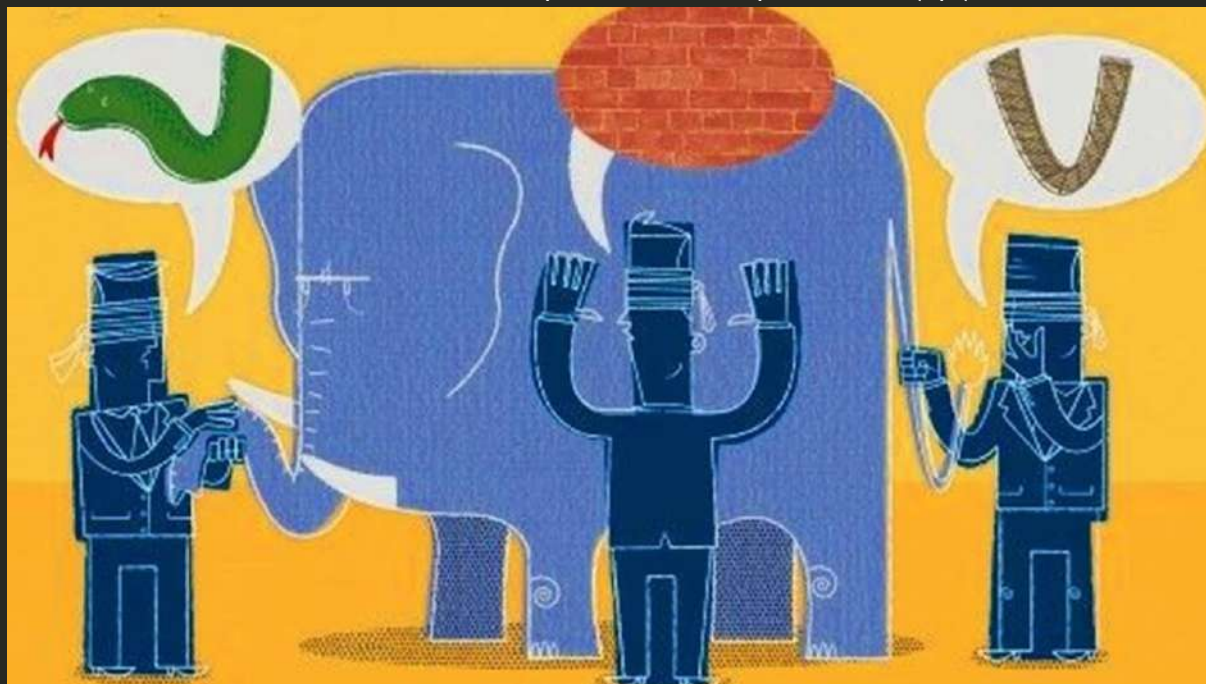


Схема Закмана

ru.wikipedia.org/wiki/Модель_Закмана

		Данные ЧТО	Функции КАК	Дислока- ция, сеть ГДЕ	Люди КТО	Время КОГДА	Мотивация ПОЧЕМУ	
Бизнес-руководители	Плани- ровщик	Список важных понятий и объектов	Список основных бизнес- процессов	Территори- альное располо- жение	Ключевые организации	Важнейшие события	Бизнес-цели и стратегии	Сфера действия (контекст)
	Владелец, менеджер	Концепту- альная модель данных	Модель бизнес- процессов	Схема логистики	Модель потока работ (workflow)	Мастер- план реализации	Бизнес-план	Модель предприятия
	Констр- уктор, архи- тектор	Логические модели данных	Архитектура приложений	Модель распреде- ленной архитектуры	Архитектура интерфейса пользо- вателя	Структура процессов	Роли и модели бизнес- правил	Модель системы
	Проекти- ровщик	Физическая модель данных	Системный проект	Технологич. архитектура	Архитектура презентации	Структуры управления	Описания бизнес- правил	Технологи- ческая (физическая) модель
	Разра- ботчик	Описание структуры данных	Програм- мный код	Сетевая архитектура	Архитектура безопас- ности	Опреде- ление временных привязок	Реализация бизнес- логики	Детали реализации
ИТ-менеджеры и разработчики		Данные	Работаю- щие программы	Сеть	Реальные люди, организа- ции	Бизнес- события	Работаю- щие бизнес- стратегии	Работающее предприятие
		Данные	Функции, Процессы	Сеть, располо- жение систем	Люди, органи- зации	Время, расписа- ния	Мотивация	

Точки зрения по Закману

Точки зрения по Закману

- Мотивация

Точки зрения по Закману

- Мотивация
- Люди

Точки зрения по Закману

- Мотивация
- Люди
- Данные

Точки зрения по Закману

- Мотивация
- Люди
- Данные
- Функции

Точки зрения по Закману

- Мотивация
- Люди
- Данные
- Функции
- Место

Точки зрения по Закману

- Мотивация
- Люди
- Данные
- Функции
- Место
- Время

Слои по Закману

Слои по Закману

- Детальное представление

Слои по Закману

- Технологическая модель
- Детальное представление

Слои по Закману

- Системная модель
- Технологическая модель
- Детальное представление

Слои по Закману

- Модель бизнеса
- Системная модель
- Технологическая модель
- Детальное представление

Слои по Закману

- Контекст
- Модель бизнеса
- Системная модель
- Технологическая модель
- Детальное представление

И при чем здесь твои примеры?

Нужно ли использовать дополнительную сущность для генерации CSS классов в Infobip?

		Данные ЧТО	Функции КАК	Дислока- ция, сеть ГДЕ	Люди КТО	Время КОГДА	Мотивация ПОЧЕМУ	
Бизнес-руководители	Плани- ровщик	Список важных понятий и объектов	Список основных бизнес- процессов	Территори- альное располо- жение	Ключевые организации	Важнейшие события	Бизнес-цели и стратегии	Сфера действия (контекст)
	Владелец, менеджер	Концепту- альная модель данных	Модель бизнес- процессов	Схема логистики	Модель потока работ (workflow)	Мастер- план реализации	Бизнес-план	Модель предприятия
	Констр- уктор, архи- тектор	Логические модели данных	Архитектура приложений	Модель распреде- ленной архитектуры	Архитектура интерфейса пользо- вателя	Структура процессов	Роли и модели бизнес- правил	Модель системы
	Проекти- ровщик	Физическая модель данных	Системный проект	Технологич. архитектура	Архитектура презентации	Структуры управления	Описания бизнес- правил	Технологи- ческая (физическая) модель
	Разра- ботчик	Описание структуры данных	Програм- мный код	Сетевая архитектура	Архитектура безопас- ности	Опреде- ление временных привязок	Реализация бизнес- логики	Детали реализации
ИТ-менеджеры и разработчики		Данные	Работаю- щие программы	Сеть	Реальные люди, организа- ции	Бизнес- события	Работаю- щие бизнес- стратегии	Работающее предприятие
		Данные	Функции, Процессы	Сеть, располо- жение систем	Люди, органи- зации	Время, расписа- ния	Мотивация	

Хоршая архитектура должна быть гибкой!

~~Хорошая архитектура должна быть гибкой!~~

Архитектура должна подходить

ВЫВОДЫ

- Архитектура – важные решения

ВЫВОДЫ

- Архитектура – важные решения
- Ищи ответы уровнем выше

ВЫВОДЫ

- Архитектура – важные решения
- Ищи ответы уровнем выше
- Лень – двигатель прогресса

СПАСИБО!

Алексей Золотых

twitter: @zolotyh

telegram: @aazolotyh