МТС ФИНТЕХ

ОТ ДИАГРАММ К ПРОЦЕССУ: КАК ВНЕДРИТЬ КУЛЬТУРУ АКТУАЛЬНОЙ АРХИТЕКТУРЫ В КОМАНДЕ

Всем привет!

Павел Германов

Руководитель направления
по архитектуре и системному
анализу в МТС Финтех

- Санкт-Петербург
- Более 15-ти лет в ИТ
- Основная специализация:
  Архитектура и системный анализ

Связаться со мной

Кому доклад будет полезен

Для специалистов, которые:

- Проектируют системы
- Работают с требованиями
- Определяют путь развития системы
- Стремятся поддерживать архитектуру в актуальном состоянии

А как дела с архитектурой у Вас?

Источник изображения: publicdomainvectors.org

Глоссарий

- C4 Model
- DSL (Domain-Specific Language) + Structurizr
- ADR (Architecture Decision Record)
- MR (Merge Request)
- Архитектурный комитет

Источник изображения: адаптированный интернет-мем

Глоссарий

Это — как «Google Maps» для вашего кода. Позволяет увидеть и весь город (систему) сверху, и отдельный переулок (компонент). Что это?

Глоссарий

Это — как «Google Maps» для вашего кода. Позволяет увидеть и весь город (систему) сверху, и отдельный переулок (компонент). Что это?

A) Jira

B) C4 Model

C) Kubernetes

Глоссарий

Это — как «Google Maps» для вашего кода. Позволяет увидеть и весь город (систему) сверху, и отдельный переулок (компонент). Что это?

A) Jira

B) C4 Model

C) Kubernetes

C4 Model

Официально: Модель для иерархического описания архитектуры программного обеспечения (Контекст, Контейнеры, Компоненты, Код)

Глоссарий

**Верите ли вы, что можно написать несколько строк текста в файле,
и он автоматически превратится в красивую диаграмму архитектуры?**

Глоссарий

Верите ли вы, что можно написать несколько строк текста в файле, и он автоматически превратится в красивую диаграмму архитектуры?

Официально: Набор инструментов, основанный на методологии C4, для создания моделей и диаграмм программной архитектуры с помощью кода

Глоссарий

Как называется документ, который отвечает на вопрос «~~Какого черта~~ Почему мы это сделали именно так?» через два года?

Глоссарий

Как называется документ, который отвечает на вопрос «~~Какого черта~~ Почему мы это сделали именно так?» через два года?

A) HR-приказ

B) Architecture Decision Record (ADR)

C) Отчёт о тестировании

Глоссарий

Как называется документ, который отвечает на вопрос «~~Какого черта~~ Почему мы это сделали именно так?» через два года?

A) HR-приказ

B) Architecture Decision Record (ADR)

C) Отчёт о тестировании

ADR (Architecture Decision Record)

Официально: Документ, фиксирующий важное архитектурное решение, его контекст, последствия и принятые альтернативы

Глоссарий

Если бы архитектура пошла в бар, что бы она взяла с собой, чтобы не устареть?

Глоссарий

Если бы архитектура пошла в бар, что бы она взяла с собой, чтобы не устареть?

A) Паспорт

B) Merge Request

C) Книгу по Python

Глоссарий

Если бы архитектура пошла в бар, что бы она взяла с собой, чтобы не устареть?

A) Паспорт

B) Merge Request

C) Книгу по Python

MR (Merge Request) / PR (Pull Request)

Официально: Механизм в системах контроля версий для предложения изменений в код и их обсуждения

Глоссарий

Что такое архитектурный комитет?

Глоссарий

Что такое архитектурный комитет?

A) Тайное общество, где решают, какие технологии запретить

B) Еженедельное чаепитие с ностальгией

C) Площадка для коллективного принятия решений

D) Процедура инициации новых разработчиков

Глоссарий

Что такое архитектурный комитет?

A) Тайное общество, где решают, какие технологии запретить

B) Еженедельное чаепитие с ностальгией

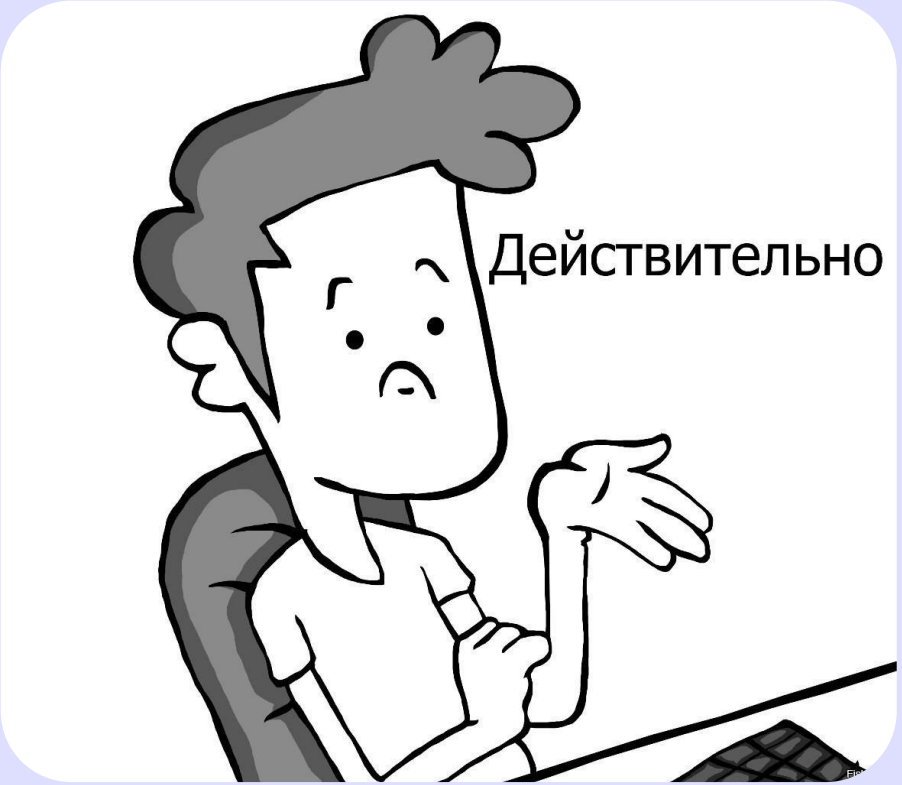
C) Площадка для коллективного принятия решений

D) Процедура инициации новых разработчиков

Официально:
Коллегиальный орган, отвечающий за выработку и контроль архитектурных решений

Зачем нам вообще нотации?

Общее понимание

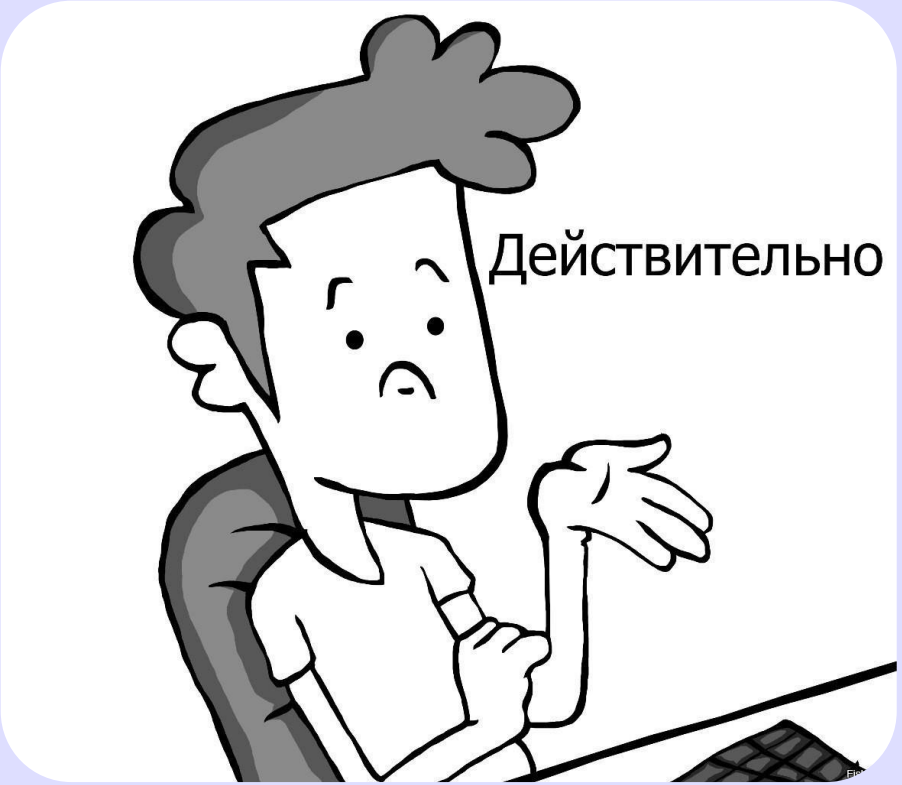


Источник изображения: адаптированный интернет-мем

Зачем нам вообще нотации?

Общее понимание

Принятие решений



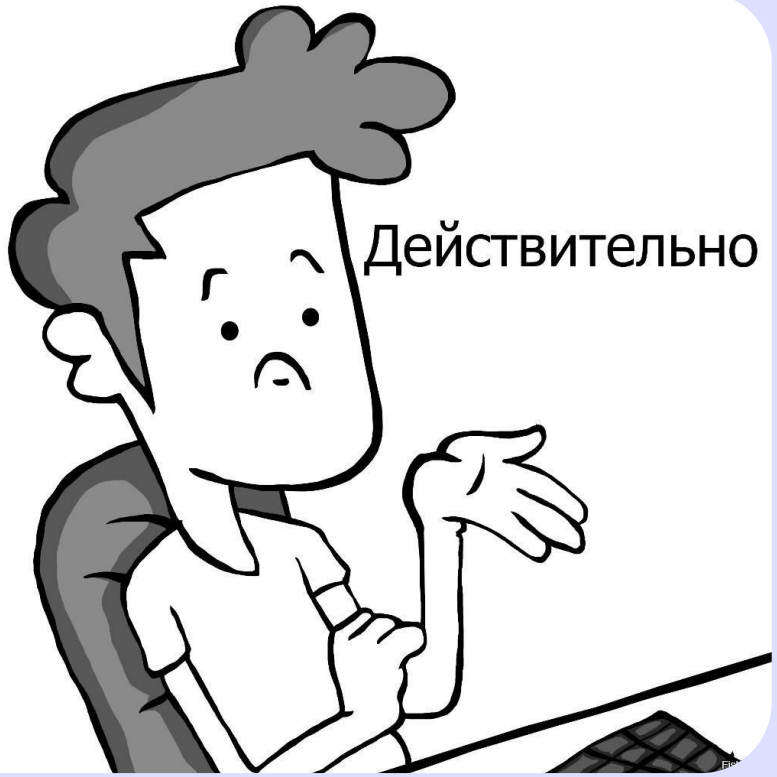
Источник изображения: адаптированный интернет-мем

Зачем нам вообще нотации?

Общее понимание

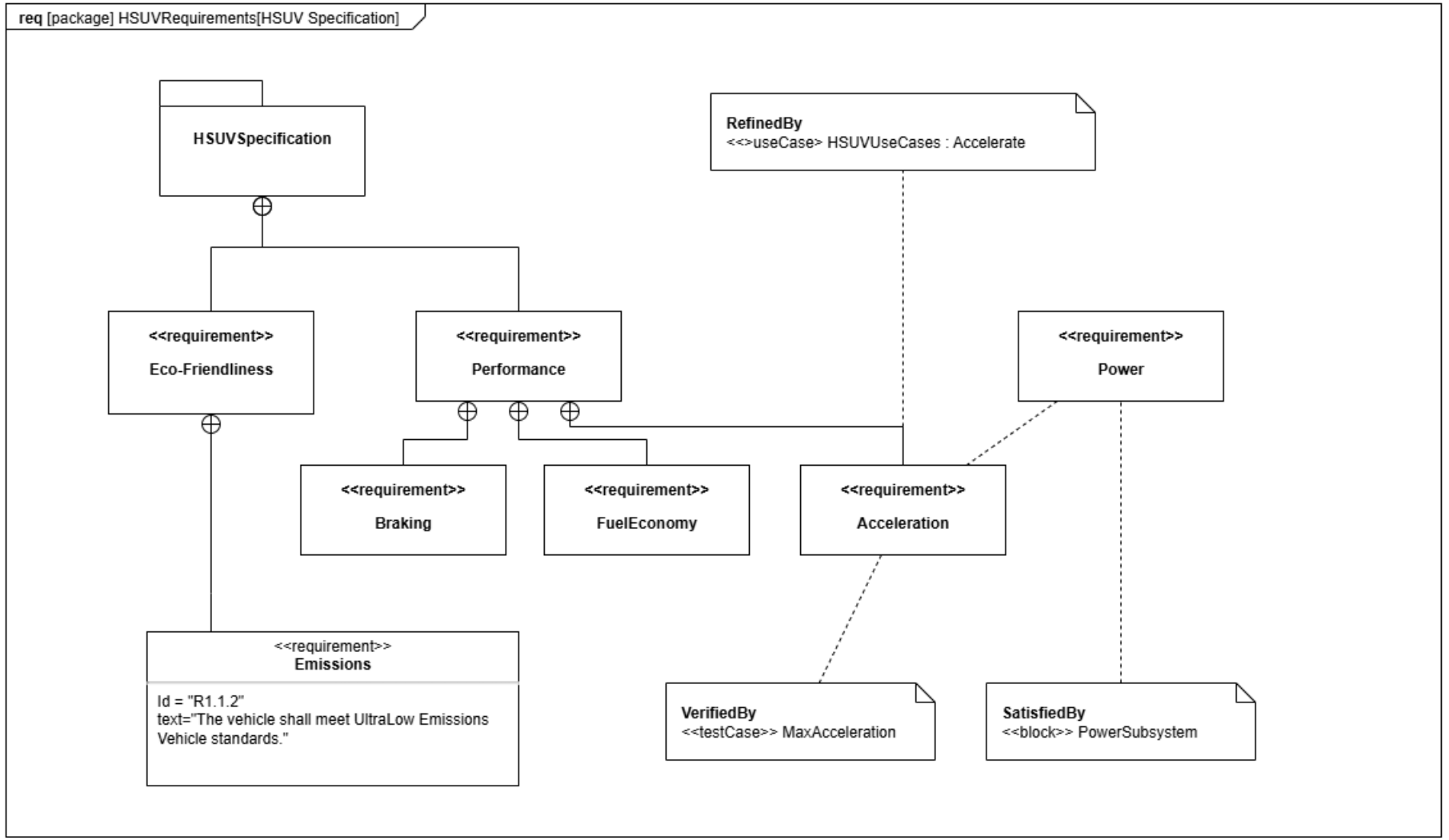
Принятие решений

Onboarding

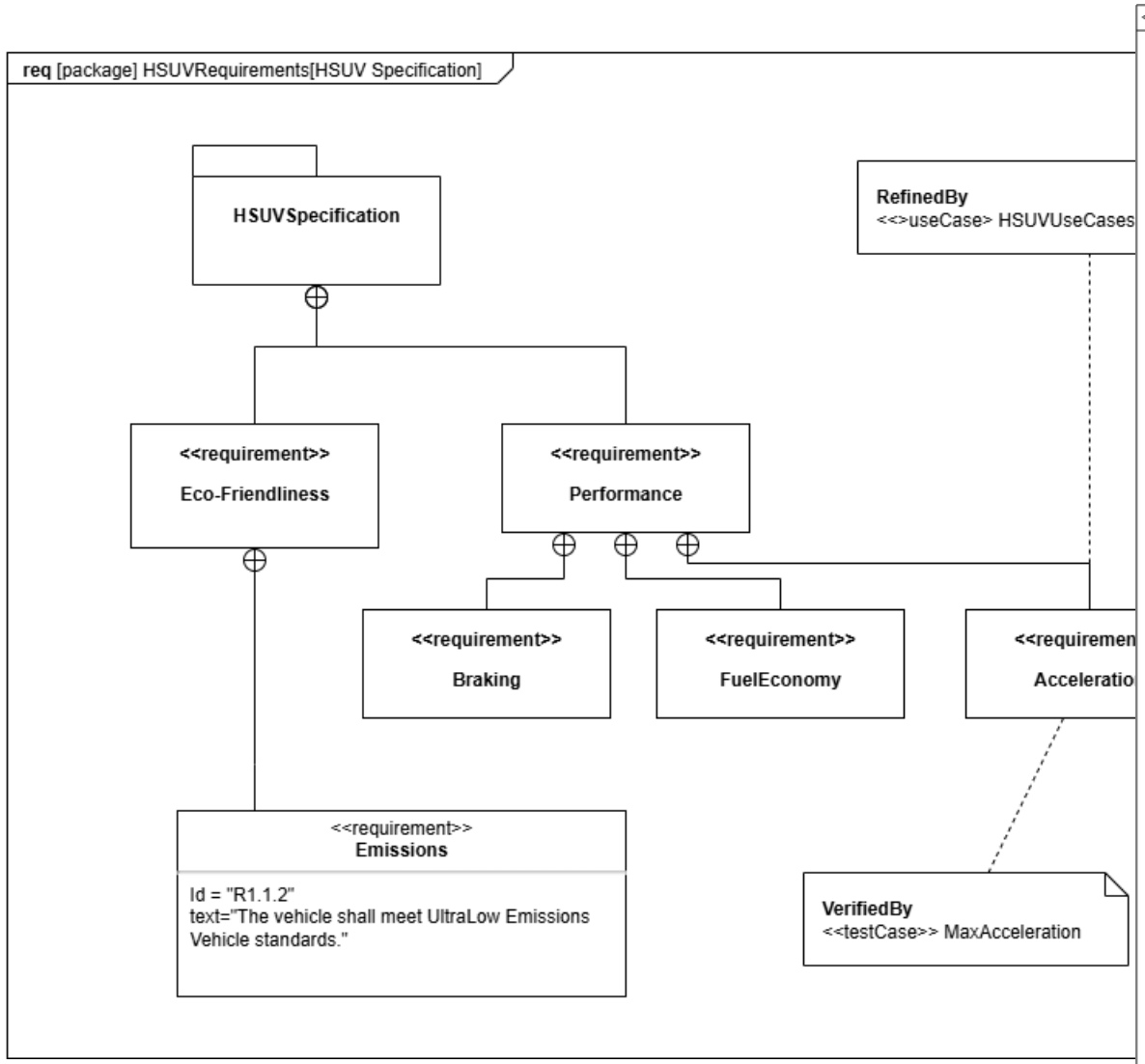


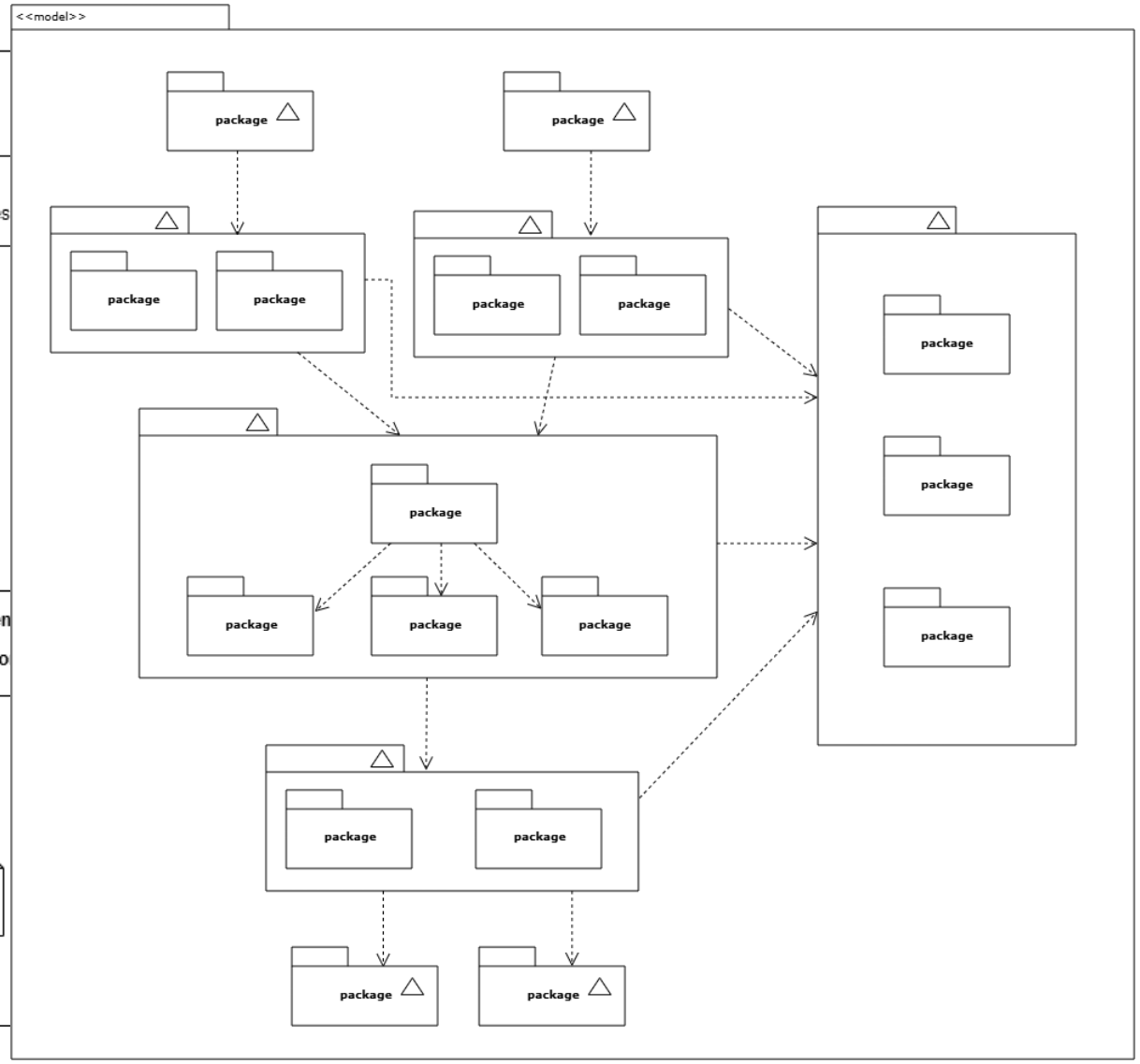
Источник изображения: адаптированный интернет-мем

Зачем нам вообще нотации?

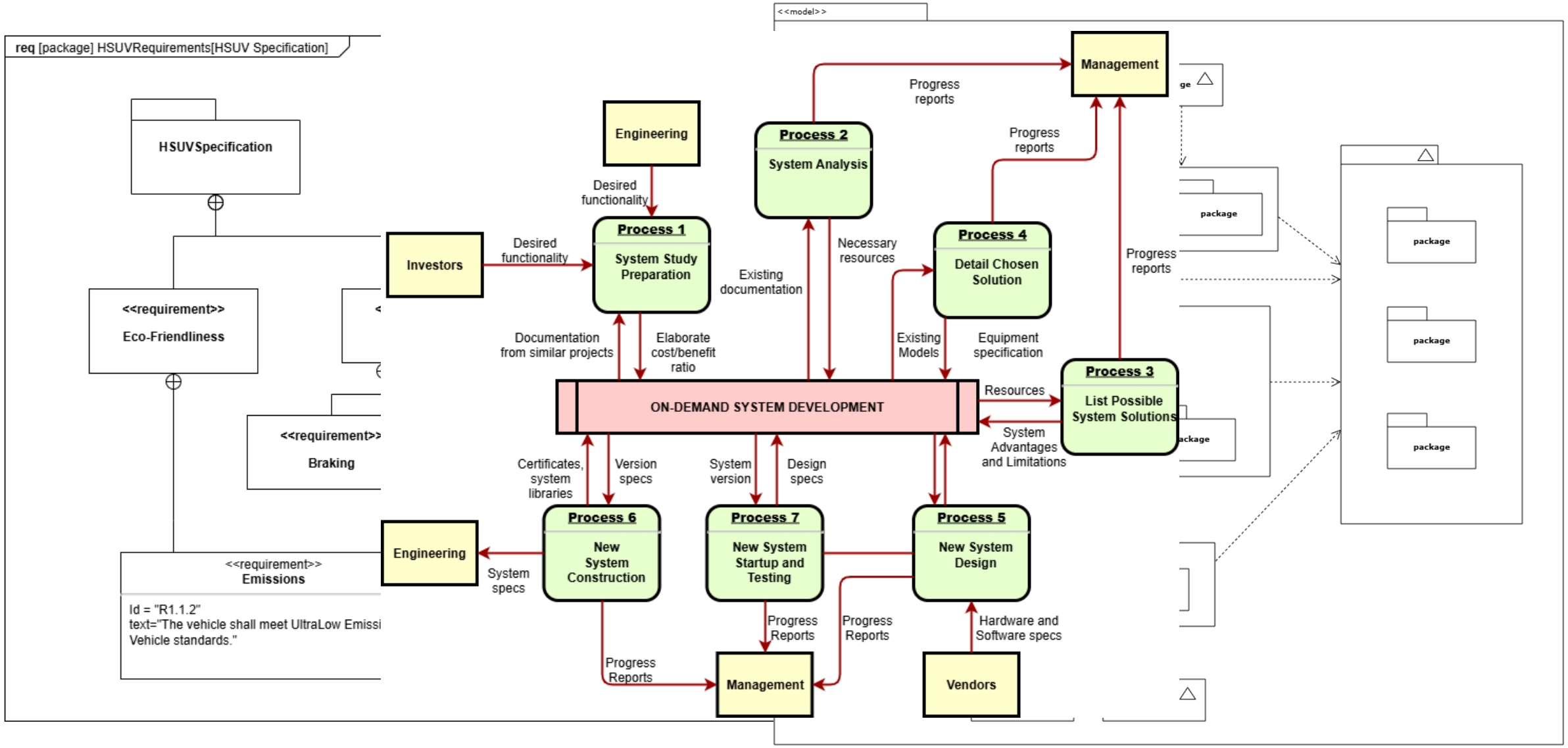


Зачем нам вообще нотации?

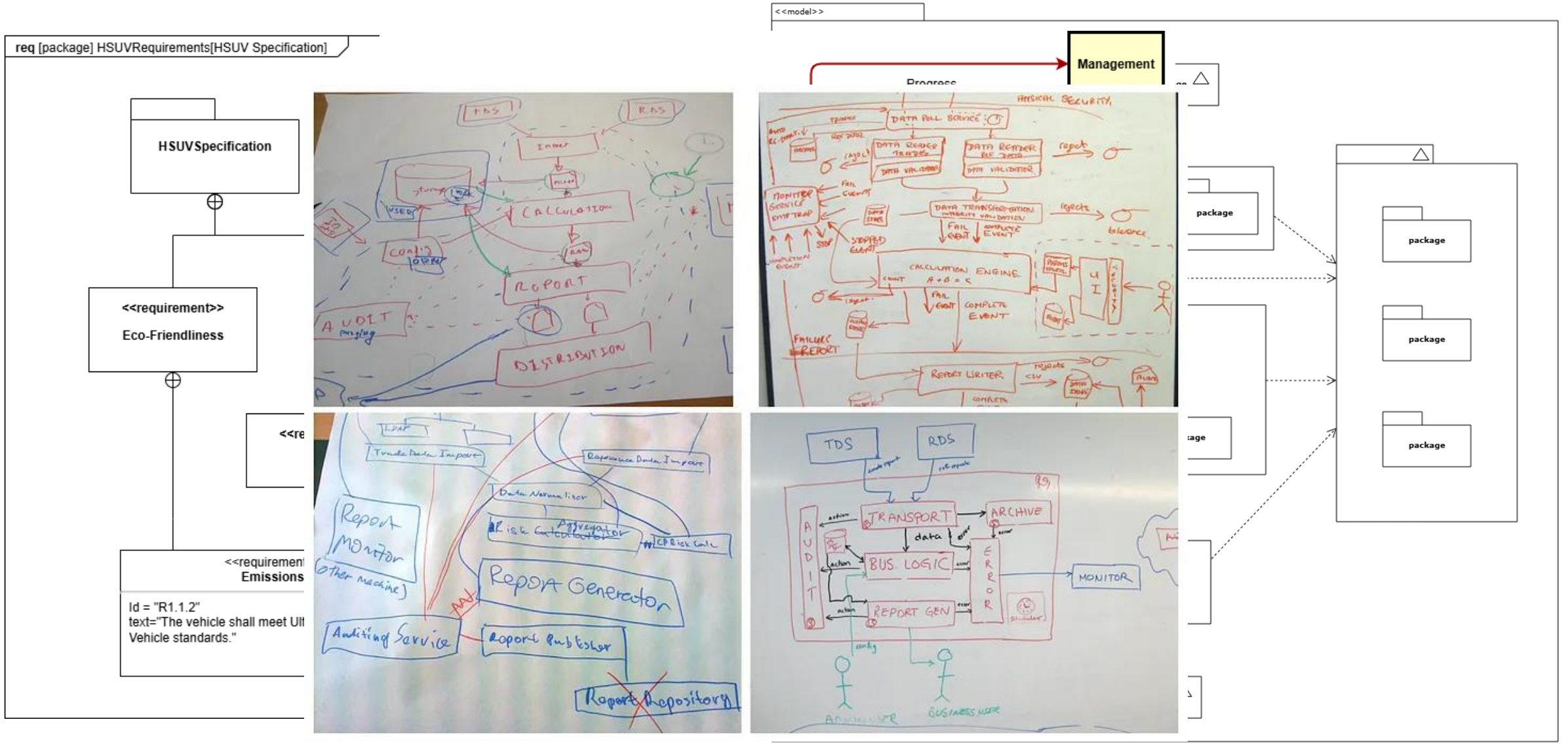




Зачем нам вообще нотации?

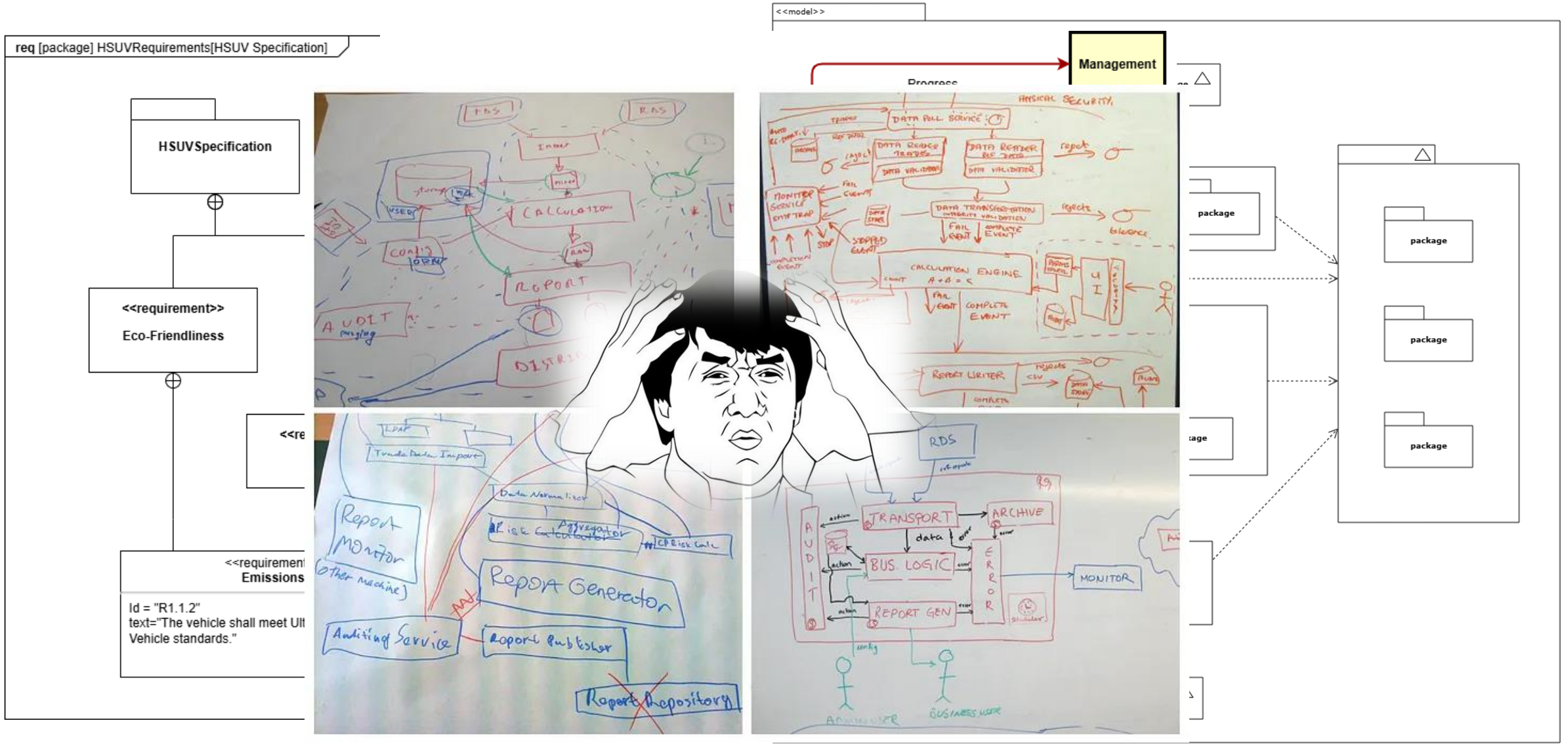


Зачем нам вообще нотации?



Источник изображения: c4model.com по лицензии Creative Commons Attribution 4.0 International License

Зачем нам вообще нотации?

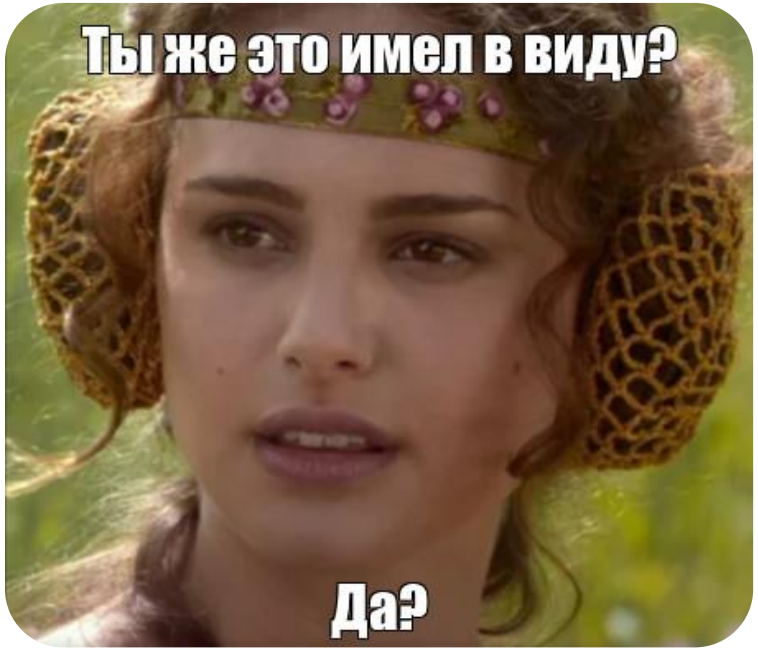


Источник изображения: c4model.com по лицензии Creative Commons Attribution 4.0 International License, адаптированный интернет-мем

Зачем нам вообще нотации?

Главная цель нотации —
убить магическое мышление

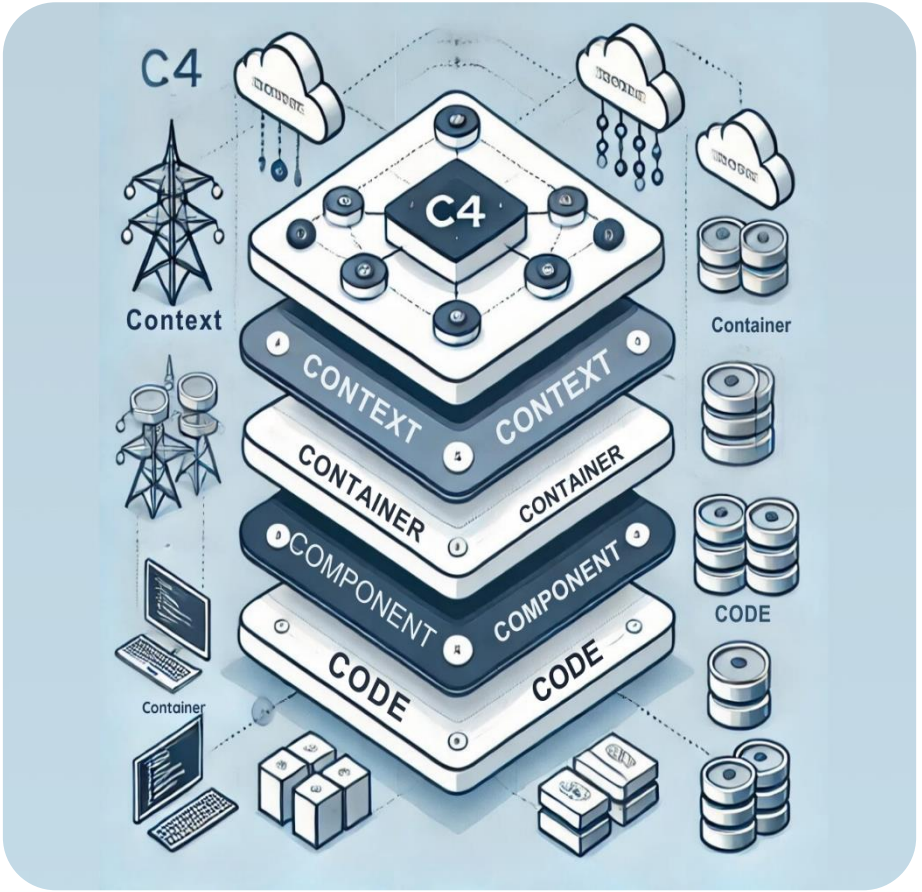
Больше никаких «я думал, ты имел ввиду...»
или «ой, а я не знал, что это отдельный сервис»



Источник изображения: адаптированный интернет-мем

C4 Model — Искусство упрощения

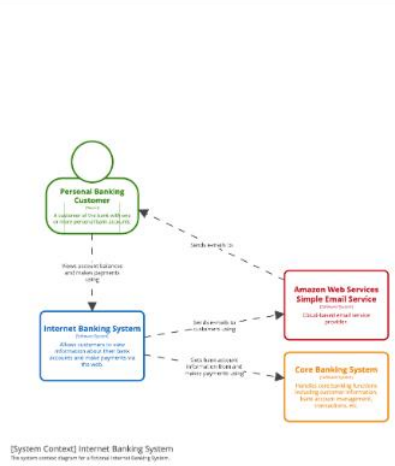
Нотация C4 Model —
один из самых популярных
и эффективных подходов



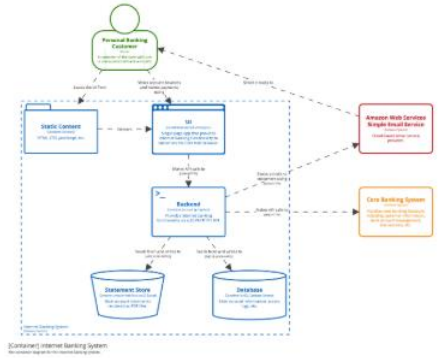
Источник изображения: blog.foxlabsdevelopers.com/how-to-create-software-architecture-diagrams-using-the-c4-model

C4 Model — Искусство упрощения

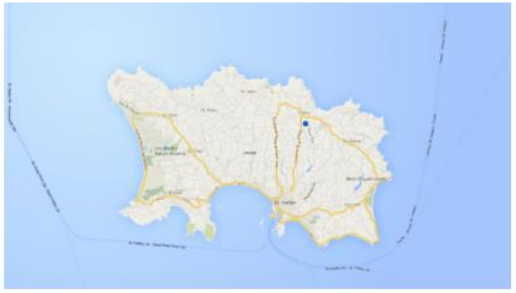
Уровень 1: Контекст

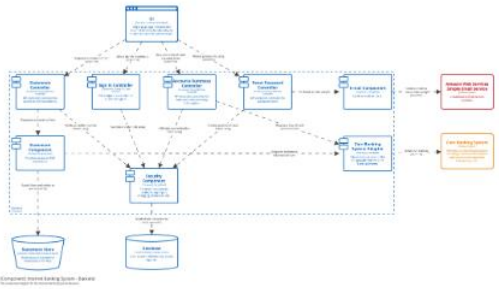


Уровень 2: Контейнеры

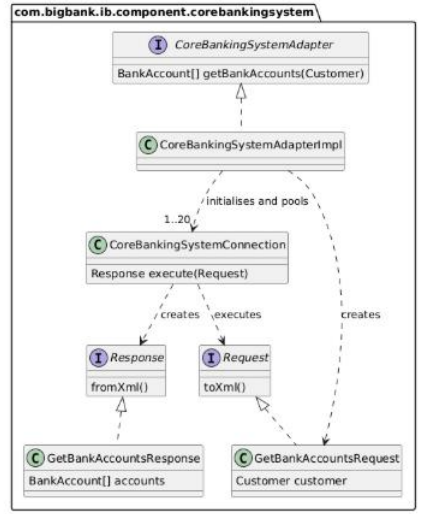


Уровень 3: Компоненты



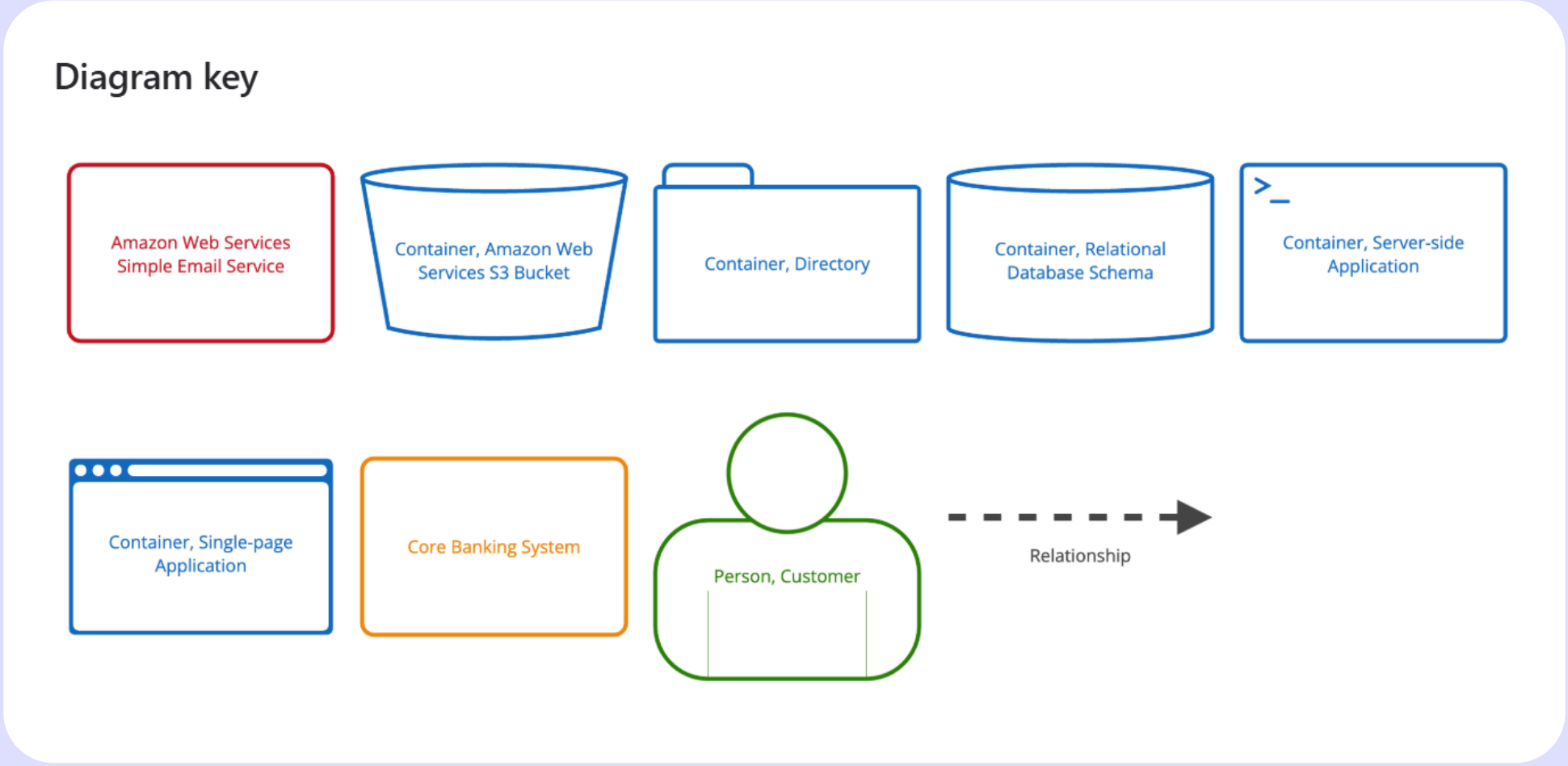


Уровень 4: Код



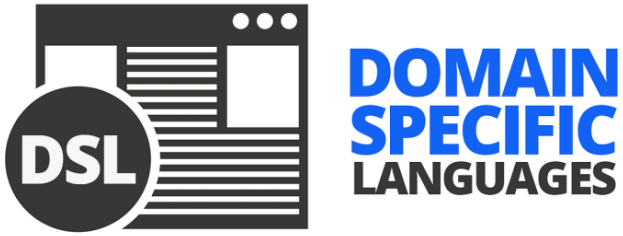
Источник изображения: c4model.com по лицензии Creative Commons Attribution 4.0 International License

C4 Model — Искусство упрощения



Источник изображения: c4model.com по лицензии Creative Commons Attribution 4.0 International License

Код, который рисует лучше художника



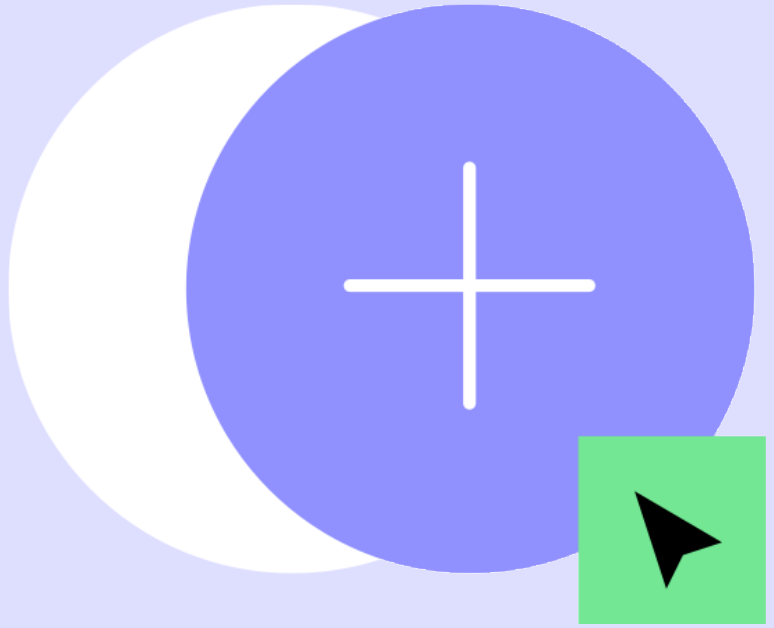
DSL (Domain Specific Languages) —
языки, специфичные для решения задач
какой-либо предметной области

- Это не Python и не Java
- Это один из языков для описания архитектуры
- Простые команды превращаются в красивые диаграммы

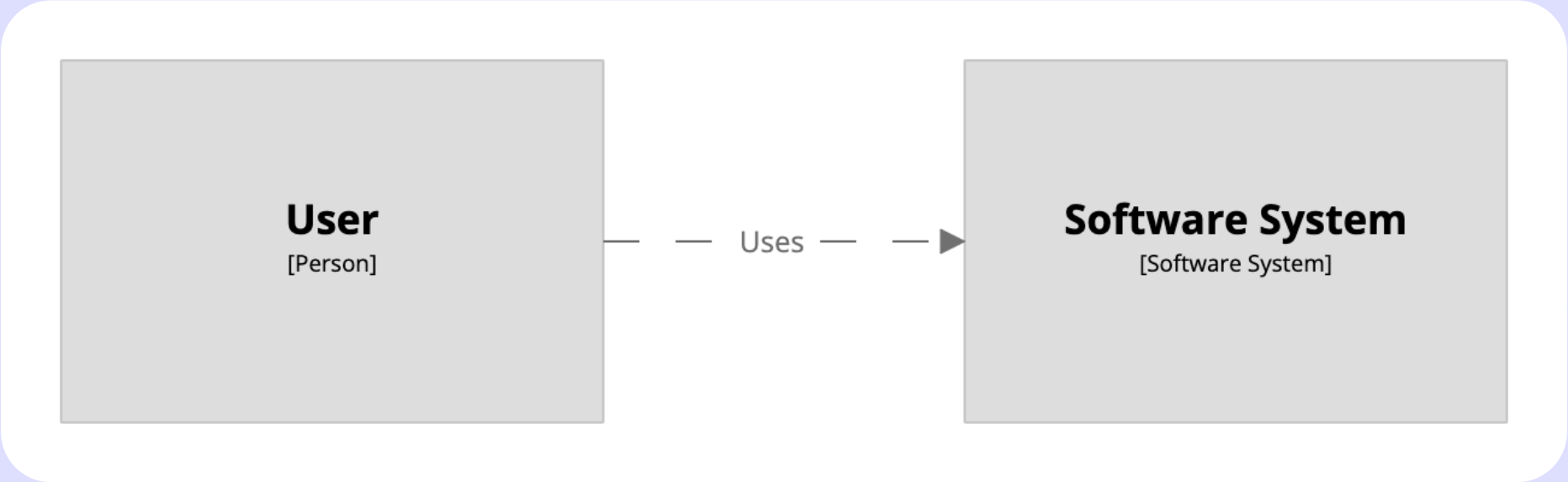
Источник изображения: digitalgadgetwave.com

Код, который рисует лучше художника

```
workspace {

    model {
        user = person "User"
        softwareSystem = softwareSystem "Software
System"

        user -> softwareSystem "Uses"
    }

    views {
        systemContext softwareSystem {
            include *
            autolayout lr
        }
    }

}
```



Код, который рисует лучше художника

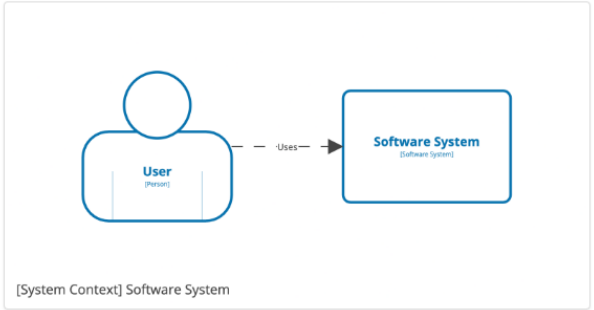


Источник изображения: structurizr.com

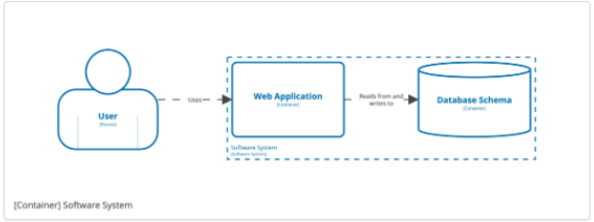
Код, который рисует лучше художника

Structurizr — волшебная палочка,
которая понимает DSL код
и молниеносно рисует по нему
диаграммы C4

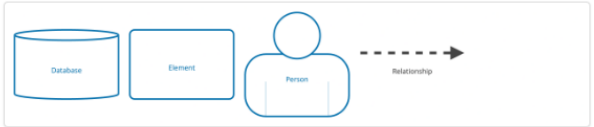
```
workspace "Name" "Description"

    !identifiers hierarchical

    model {
        u = person "User"
        ss = softwareSystem "Software System" {
            wa = container "Web Application"
            db = container "Database Schema" {
                tags "Database"
            }
        }

        u -> ss "Uses"
        u -> ss.wa "Uses"
        ss.wa -> ss.db "Reads from and writes to"
    }

    views {
        systemContext ss "Diagram1" {
            include *
            autolayout lr
        }

        container ss "Diagram2" {
            include *
            autolayout lr
        }

        // styling...
    }

}
```



[System Context] Software System

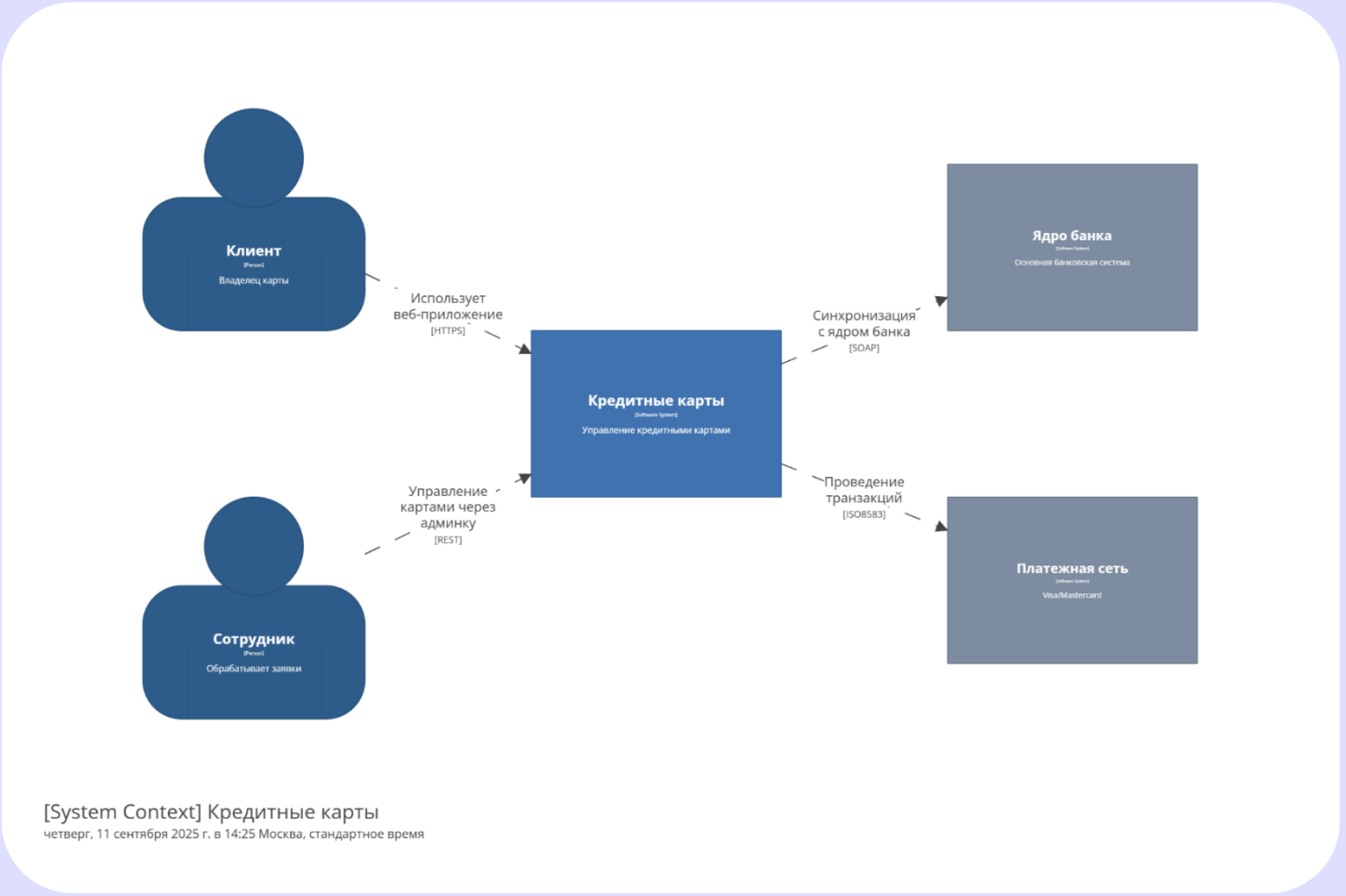


[Container] Software System

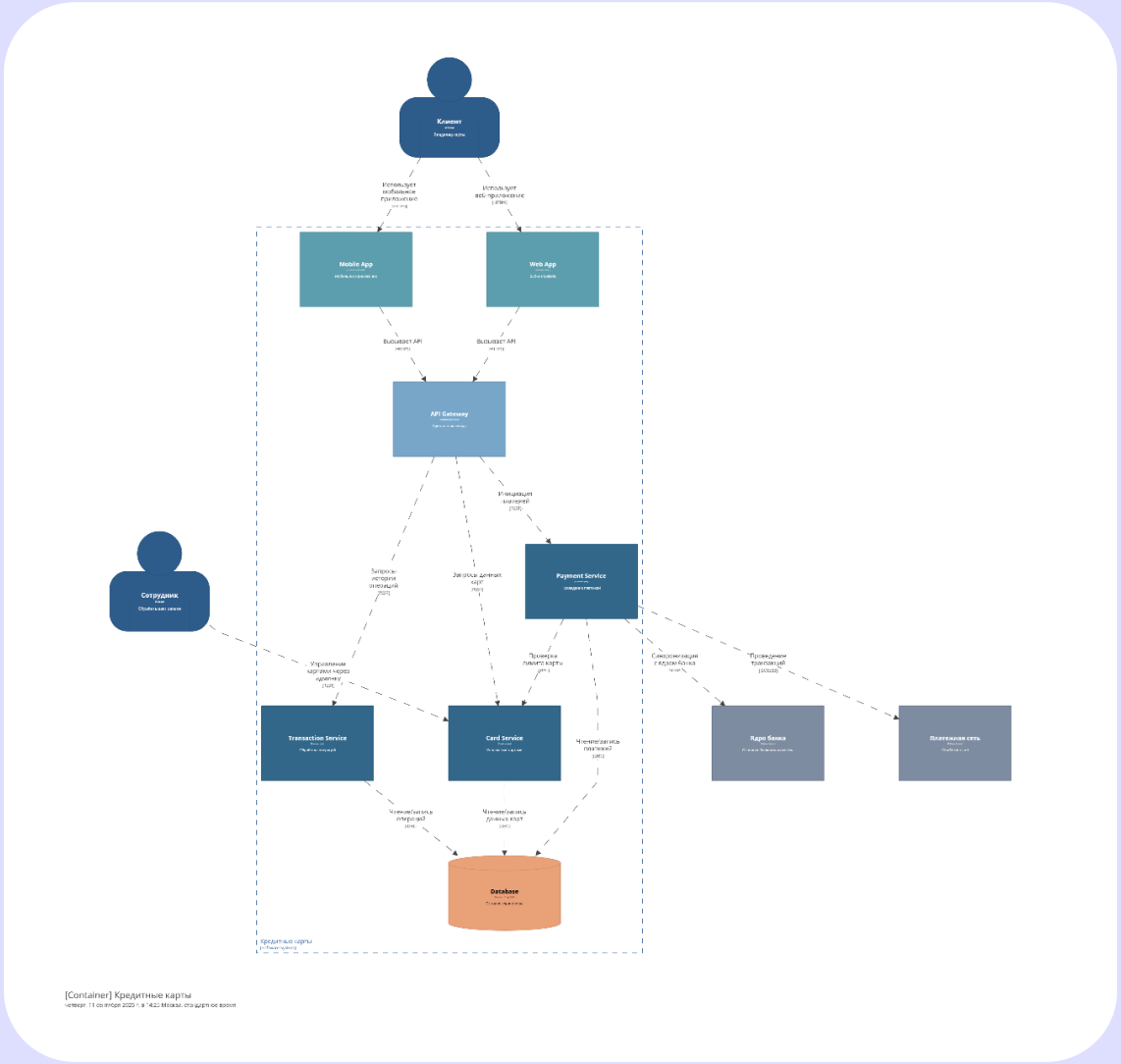


Источник изображения: structurizr.com

Код, который рисует лучше художника



Код, который рисует лучше художника



Код, который рисует лучше художника

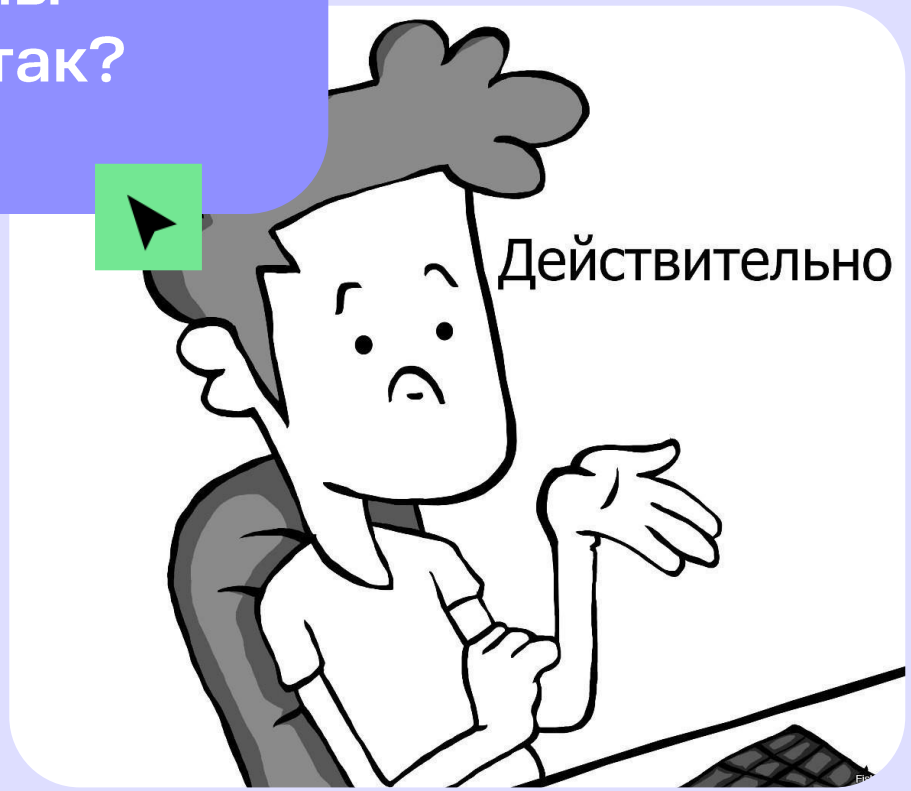
Главный бонус — архитектура становится машиночитаемой, её можно:

- Версионировать
- Проверять на ошибки
- Встраивать в CI/CD процессы

Источник изображения: digitalgadgetwave.com, structurizr.com

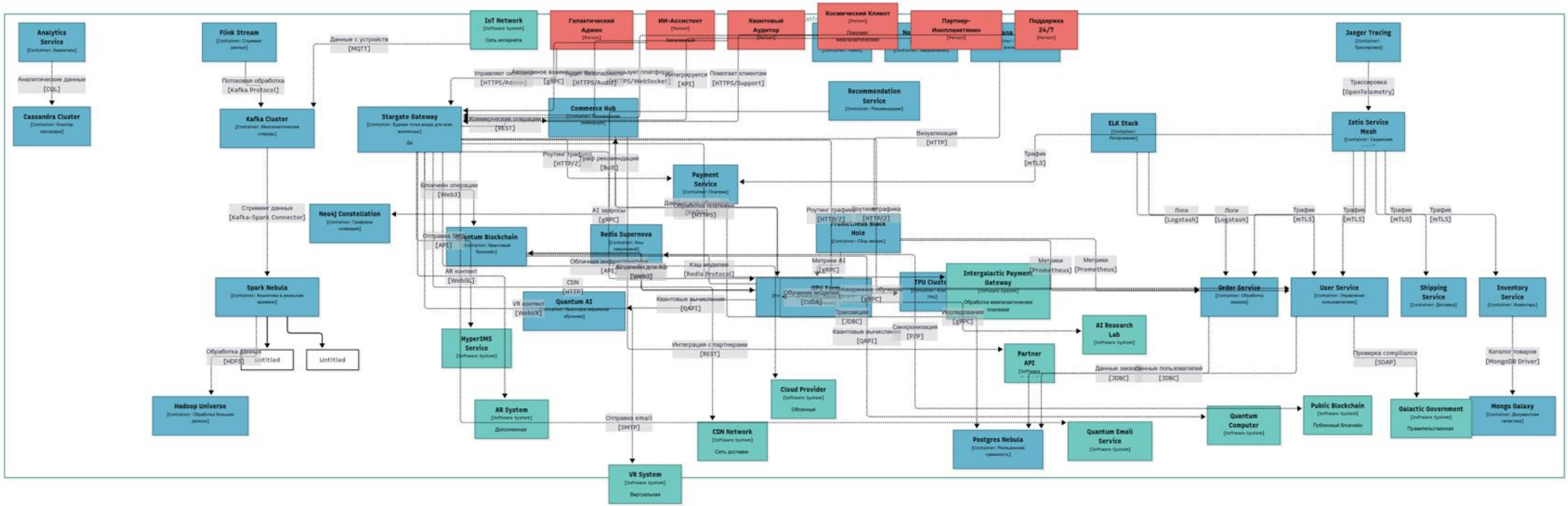
Капсула времени для ваших решений

Зачем и почему мы сделали именно так?



Источник изображения: адаптированный интернет-мем

Капсула времени для ваших решений



Капсула времени для ваших решений

ADR — это короткий документ, который описывает принятое архитектурное решение, контекст, последствия и причины

В какой ситуации мы были? (Контекст)

Капсула времени для ваших решений

ADR — это короткий документ, который описывает принятое архитектурное решение, контекст, последствия и причины

В какой ситуации мы были? (Контекст)

Что мы решили? (Решение)

Капсула времени для ваших решений

ADR — это короткий документ, который описывает принятое архитектурное решение, контекст, последствия и причины

В какой ситуации мы были? (Контекст)

Что мы решили? (Решение)

Почему мы решили именно так? (Обоснование)

Капсула времени для ваших решений

ADR — это короткий документ, который описывает принятое архитектурное решение, контекст, последствия и причины

В какой ситуации мы были? (Контекст)

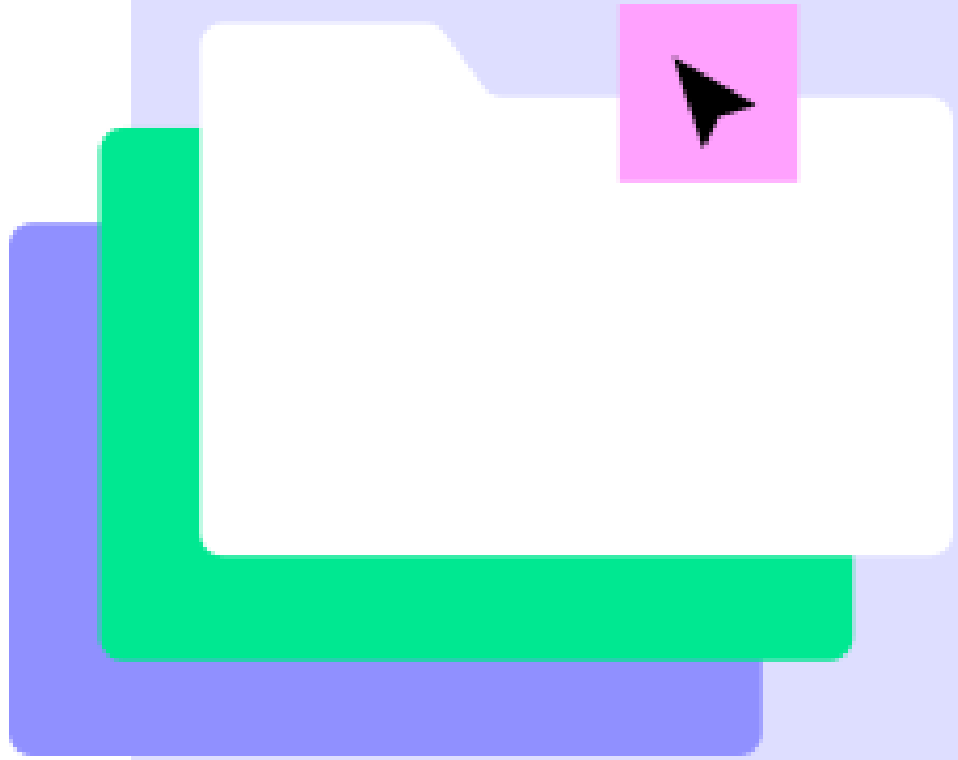
Что мы решили? (Решение)

Почему мы решили именно так? (Обоснование)

А что мы отклонили? (Рассмотренные альтернативы)

Капсула времени для ваших решений

1. Заголовок
2. Статус
3. Контекст
4. Рассмотренные варианты
5. Решение
6. Последствия
7. Участники решения



Капсула времени для ваших решений

Расширенные и часто используемые разделы:

- Ответственные
- Ссылки на реализацию
- Связанные решения
- Примечания

Капсула времени для ваших решений

ADR 001:
Выбор базы данных

Статус:
Принято

Контекст:
- Нужна надежная БД для хранения структурированных данных пользователей
- Требуются транзакции и сложные запросы

Решение:
Используем PostgreSQL

Обоснование:
- Полная поддержка SQL и ACID
- Надежность и большое сообщество
- Опыт команды

Последствия:
- Надежность, богатый функционал.
- Сложнее масштабируется, чем NoSQL

Ни один байт не уйдет без review

Git — это распределенная система контроля версий,
которая позволяет отслеживать все изменения
в файлах проекта, управлять их различными версиями

Источник изображения: commons.wikimedia.org по лицензии Creative Commons Attribution 4.0 International license

Ни один байт не уйдет без review

Git — это распределенная система контроля версий, которая позволяет отслеживать все изменения в файлах проекта, управлять их различными версиями

Git — наш главный хранитель времени. Каждое изменение архитектуры (как и кода) фиксируется в коммите

Источник изображения: commons.wikimedia.org по лицензии Creative Commons Attribution 4.0 International license

Ни один байт не уйдет без review

Merge Request (MR) — это механизм, используемый в системах управления версиями, таких как GitLab, для внесения и проверки изменений в коде из одной ветки в другую

Источник изображения: commons.wikimedia.org по лицензии MIT

Ни один байт не уйдет без review

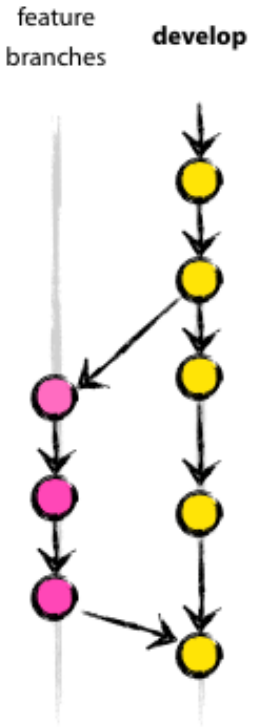
Merge Request (MR) — это механизм, используемый в системах управления версиями, таких как GitLab, для внесения и проверки изменений в коде из одной ветки в другую

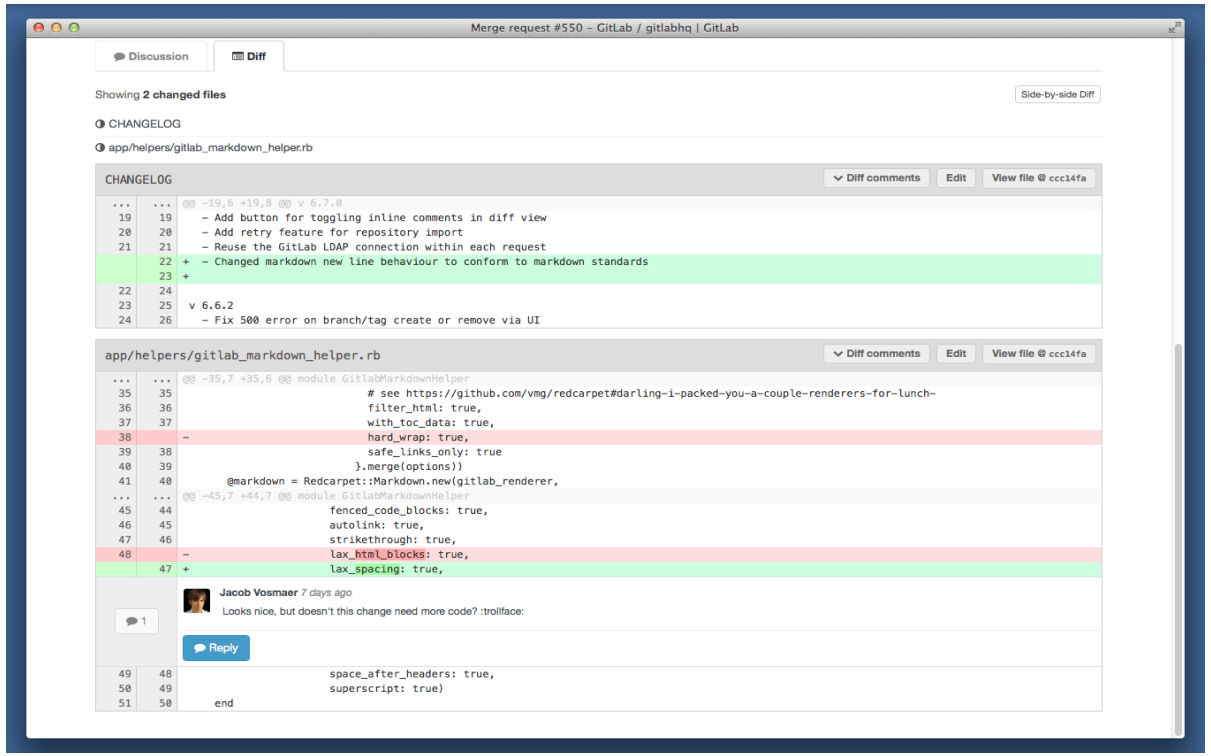
Merge Request (MR) — это «шлюз качества»

- Хочешь изменить архитектуру? — Создай MR
- Опиши причину
- Коллеги оставляют правки и комментарии

Источник изображения: commons.wikimedia.org по лицензии MIT

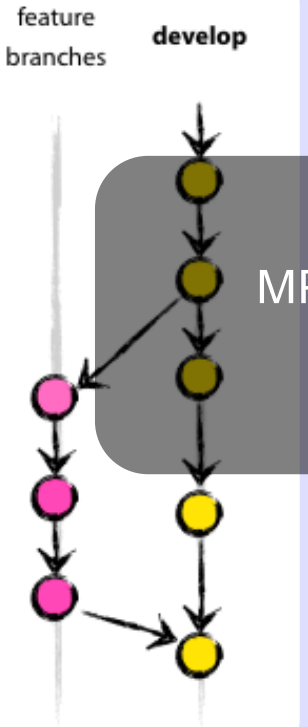
Ни один байт не уйдет без review

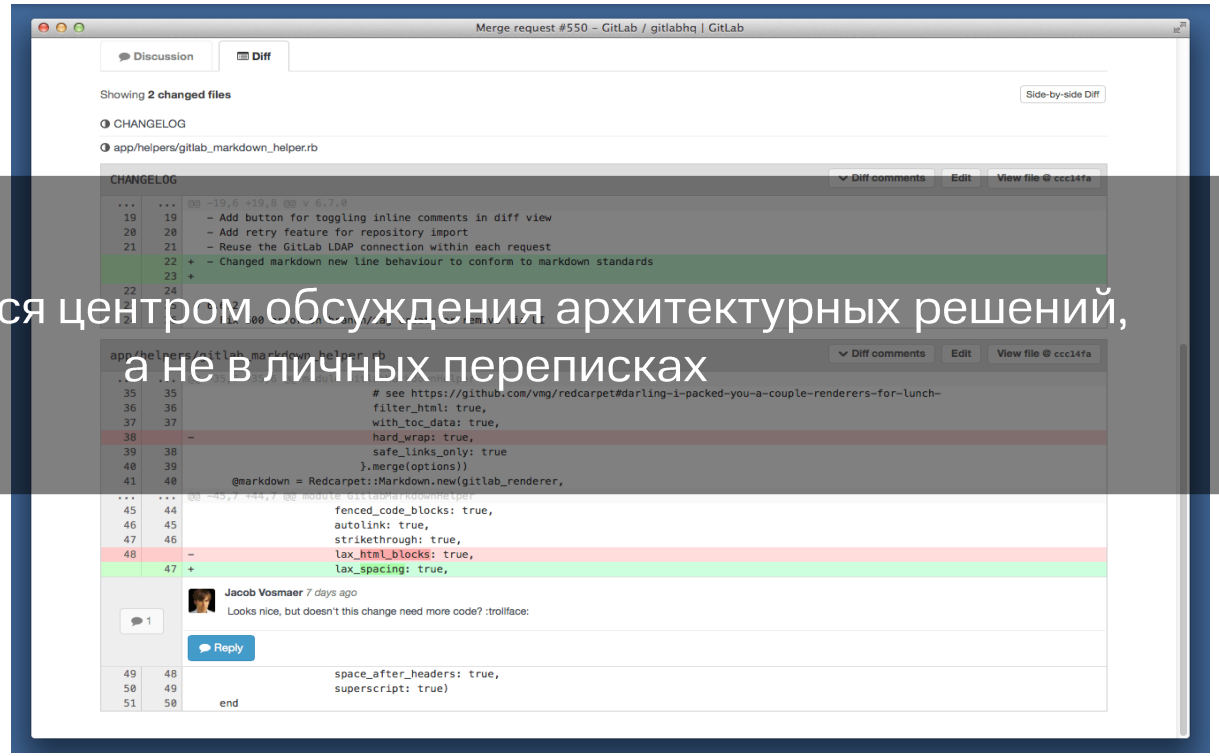




Источник изображения: Vincent Driessen | Original blog post | License: Creative Commons BY-SA, gitlab.com | License: MIT

Ни один байт не уйдет без review



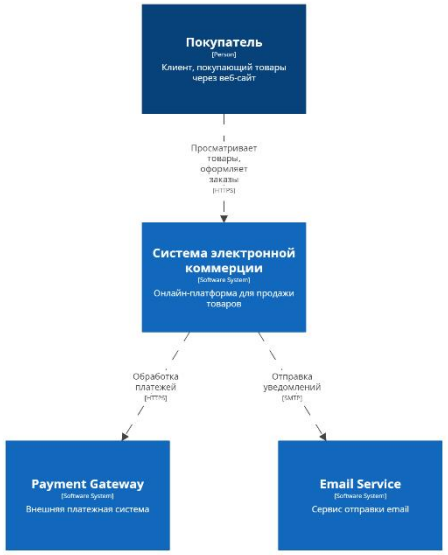


MR становится центром обсуждения архитектурных решений, а не в личных переписках

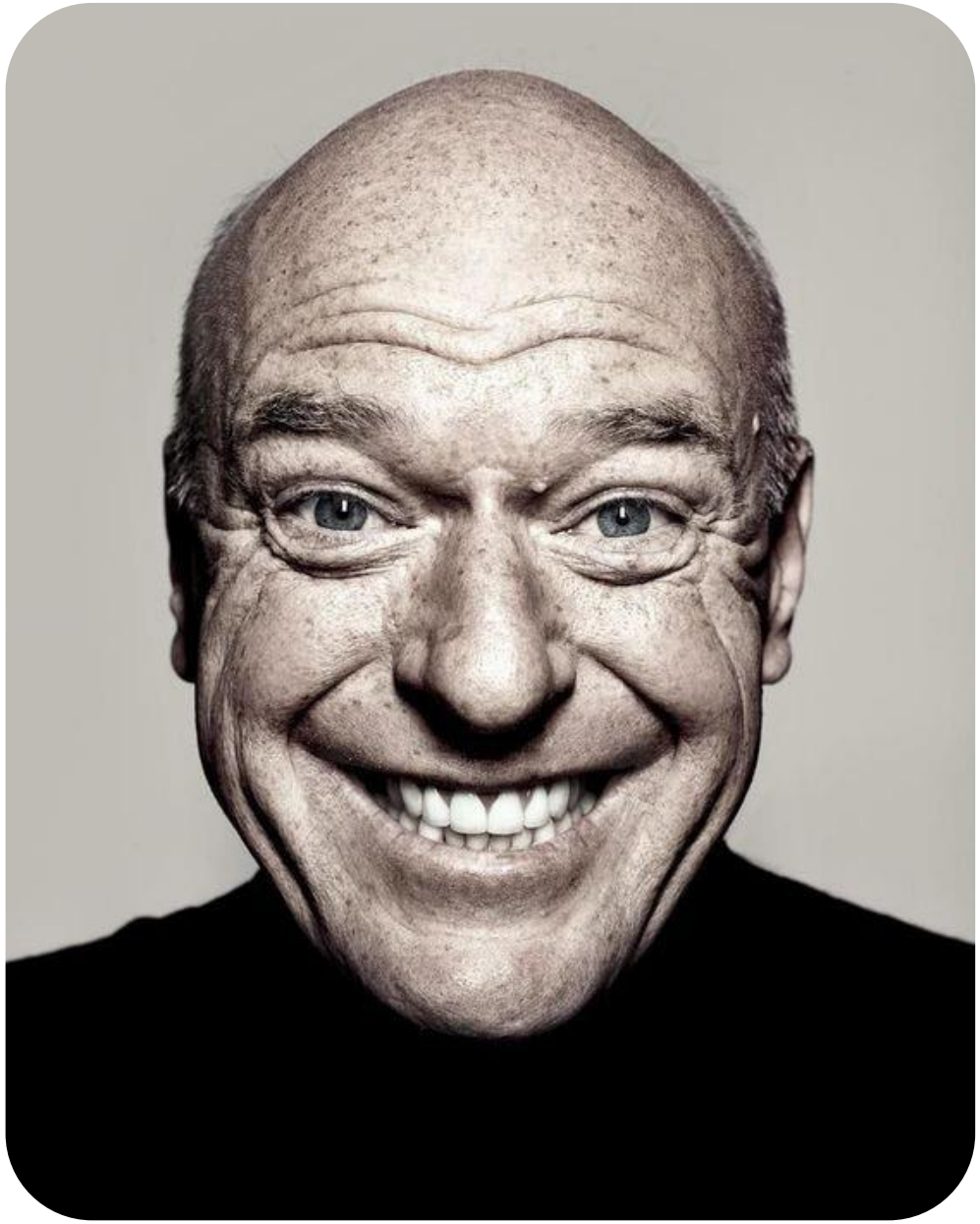
Источник изображения: Vincent Driessen | Original blog post | License: Creative Commons BY-SA, gitlab.com | License: MIT

«Архитектура-зомби»

Она выглядит как живая...



[System Context] Система электронной коммерции
пятница, 19 сентября 2025 г. в 22:38 Москва, стандартное время

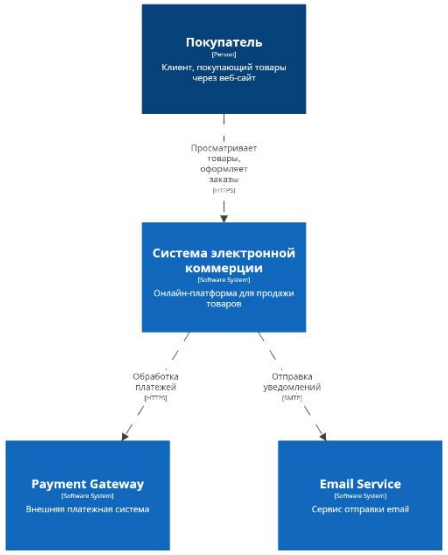


Источник изображения: адаптированный интернет-мем

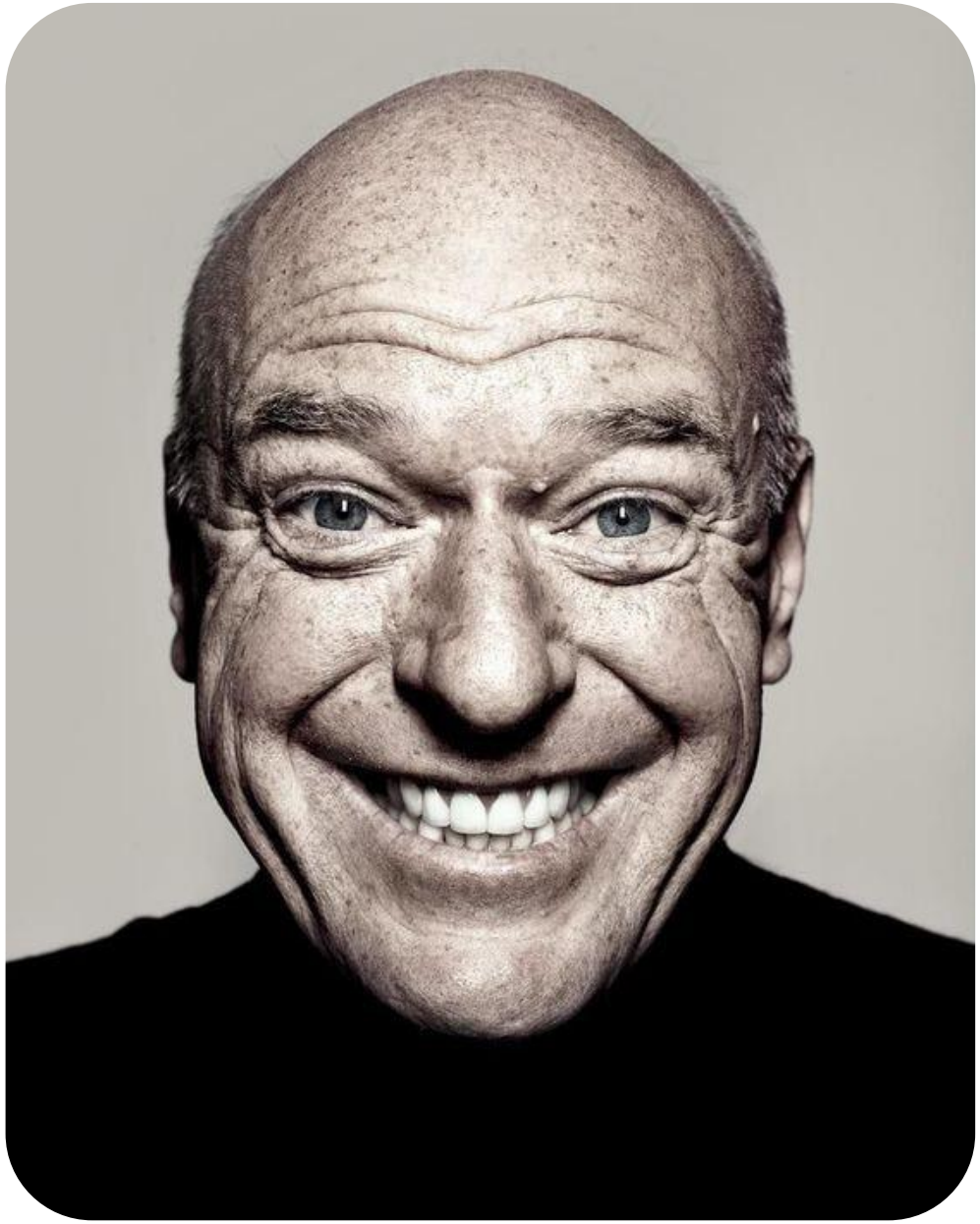
«Архитектура-зомби»

Она выглядит как живая...

- Красивые диаграммы, нарисованные в прошлом квартале



[System Context] Система электронной коммерции
пятница, 19 сентября 2025 г. в 22:38 Москва, стандартное время

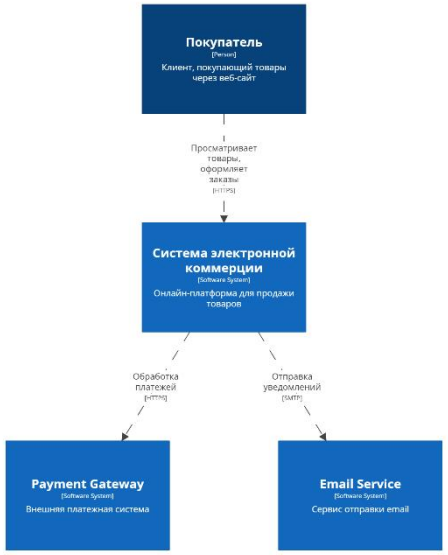


Источник изображения: адаптированный интернет-мем

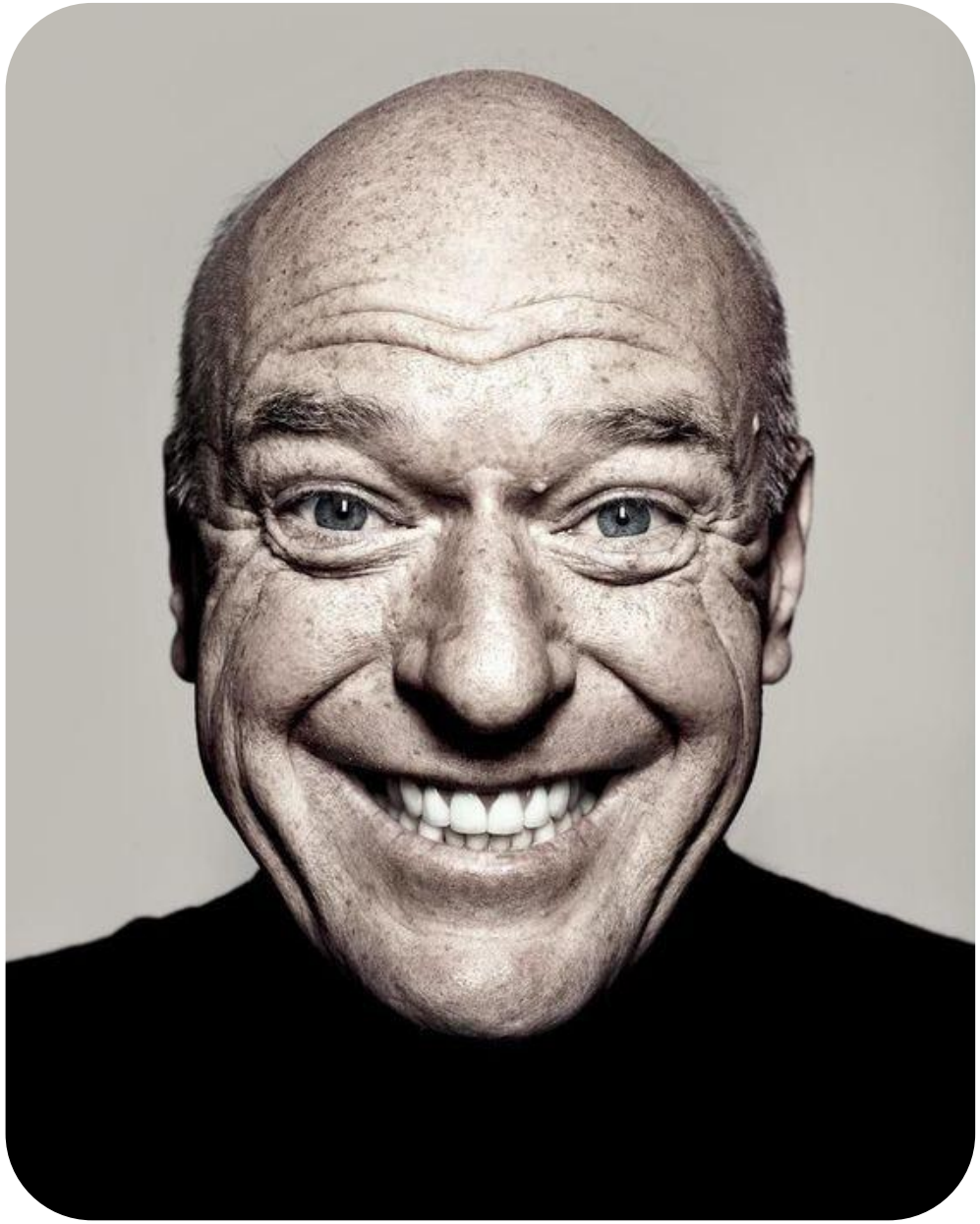
«Архитектура-зомби»

Она выглядит как живая...

- Красивые диаграммы, нарисованные в прошлом квартале
- Лежит в Конфлюенсе и создает видимость порядка



[System Context] Система электронной коммерции
пятница, 19 сентября 2025 г. в 22:38 Москва, стандартное время

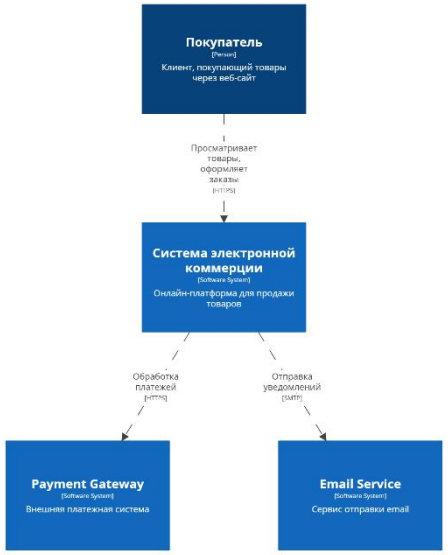


Источник изображения: адаптированный интернет-мем

«Архитектура-зомби»

Она выглядит как живая...

- Красивые диаграммы, нарисованные в прошлом квартале
- Лежит в Конфлюенсе и создает видимость порядка



[System Context] Система электронной коммерции
пятница, 19 сентября 2025 г. в 22:38 Москва, стандартное время

... но это не так

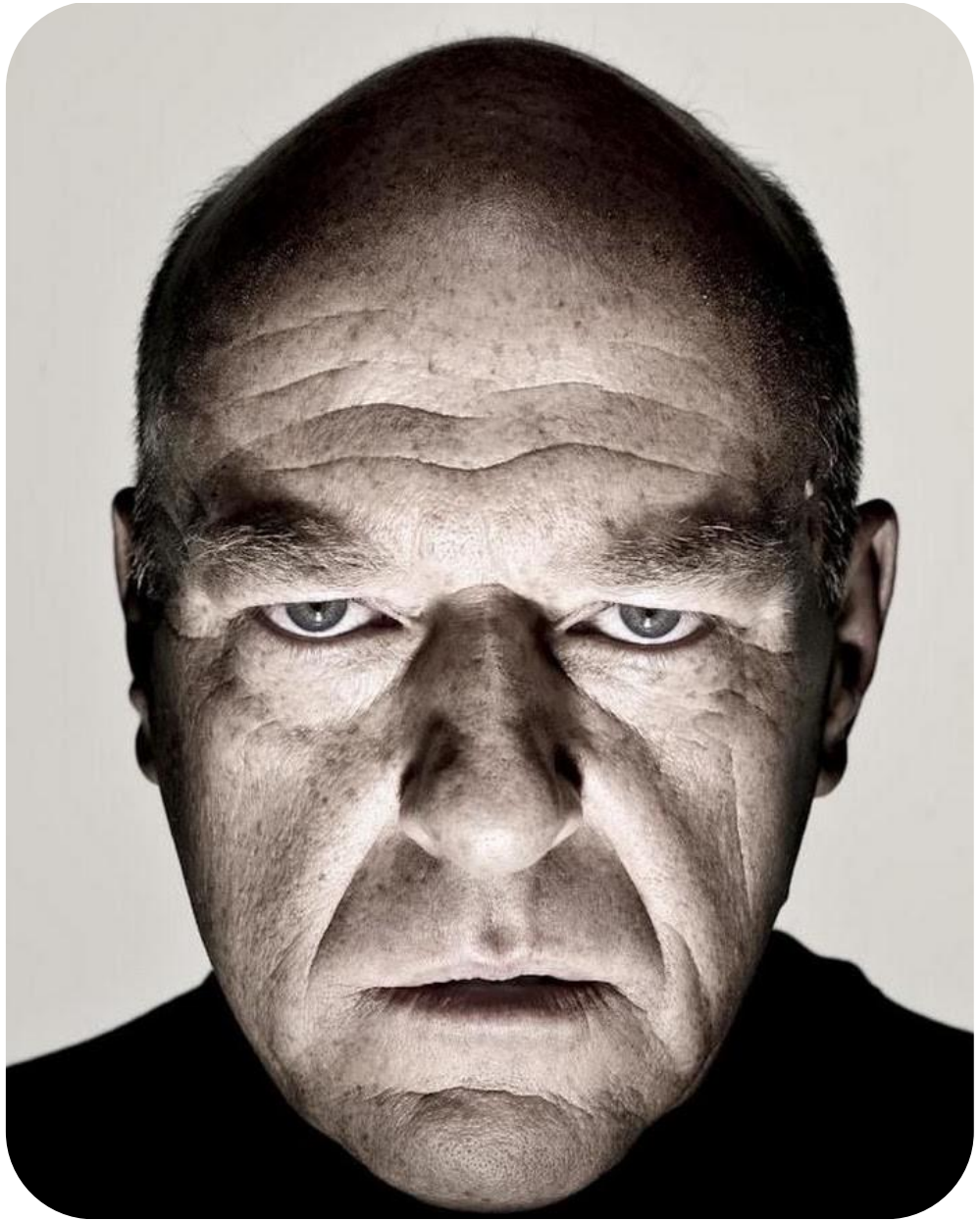
Источник изображения: адаптированный интернет-мем

«Архитектура-зомби»

Симптомы заражения проекта зомби-архитектурой

Вы часто слышите фразы:

- «Диаграмма не актуальна, но там в целом всё так..»



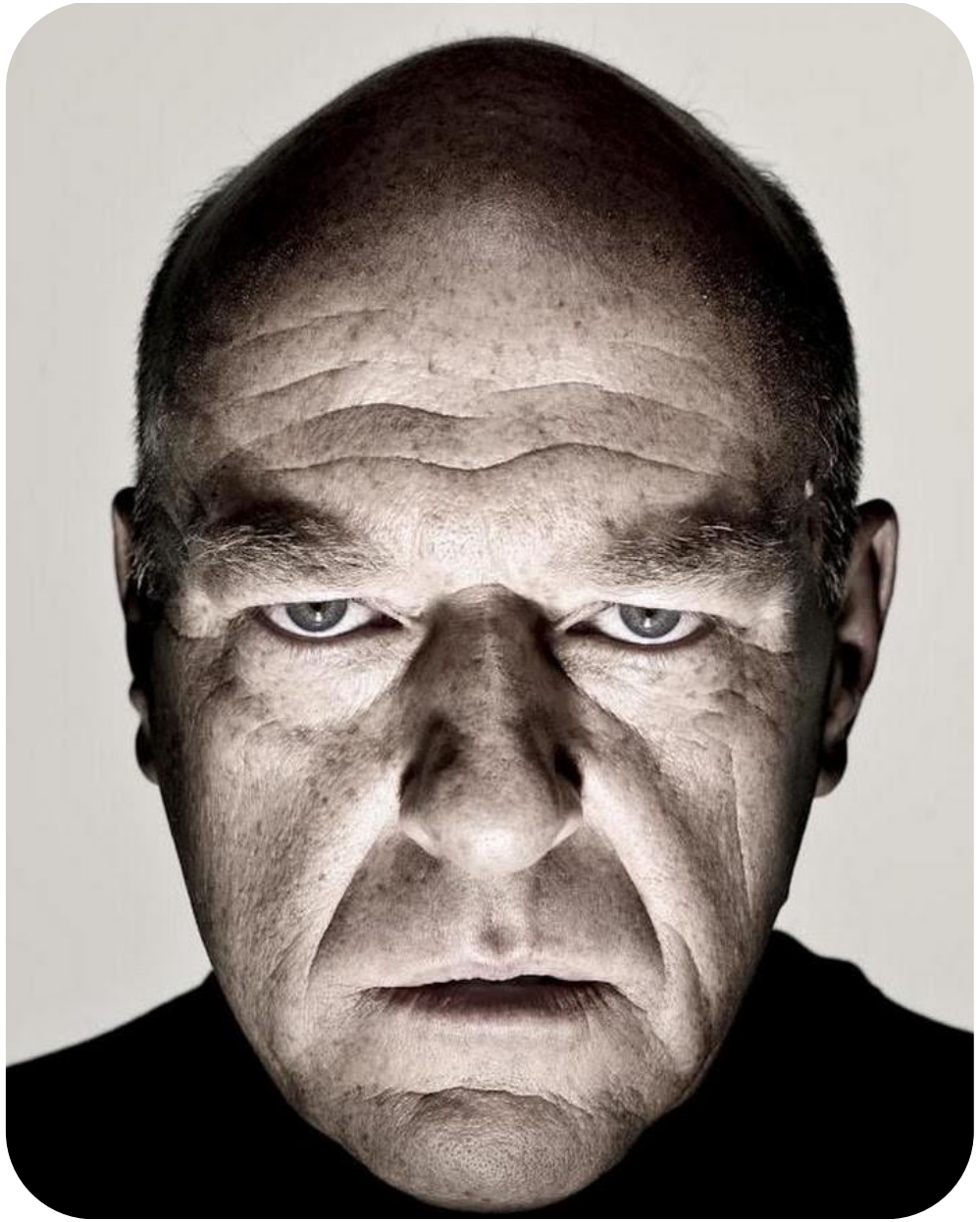
Источник изображения: адаптированный интернет-мем

«Архитектура-зомби»

Симптомы заражения проекта зомби-архитектурой

Вы часто слышите фразы:

- «Диаграмма не актуальна,
  но там в целом всё так..»
- «Лучше спросить у разрабов,
  в доке там не совсем так...»



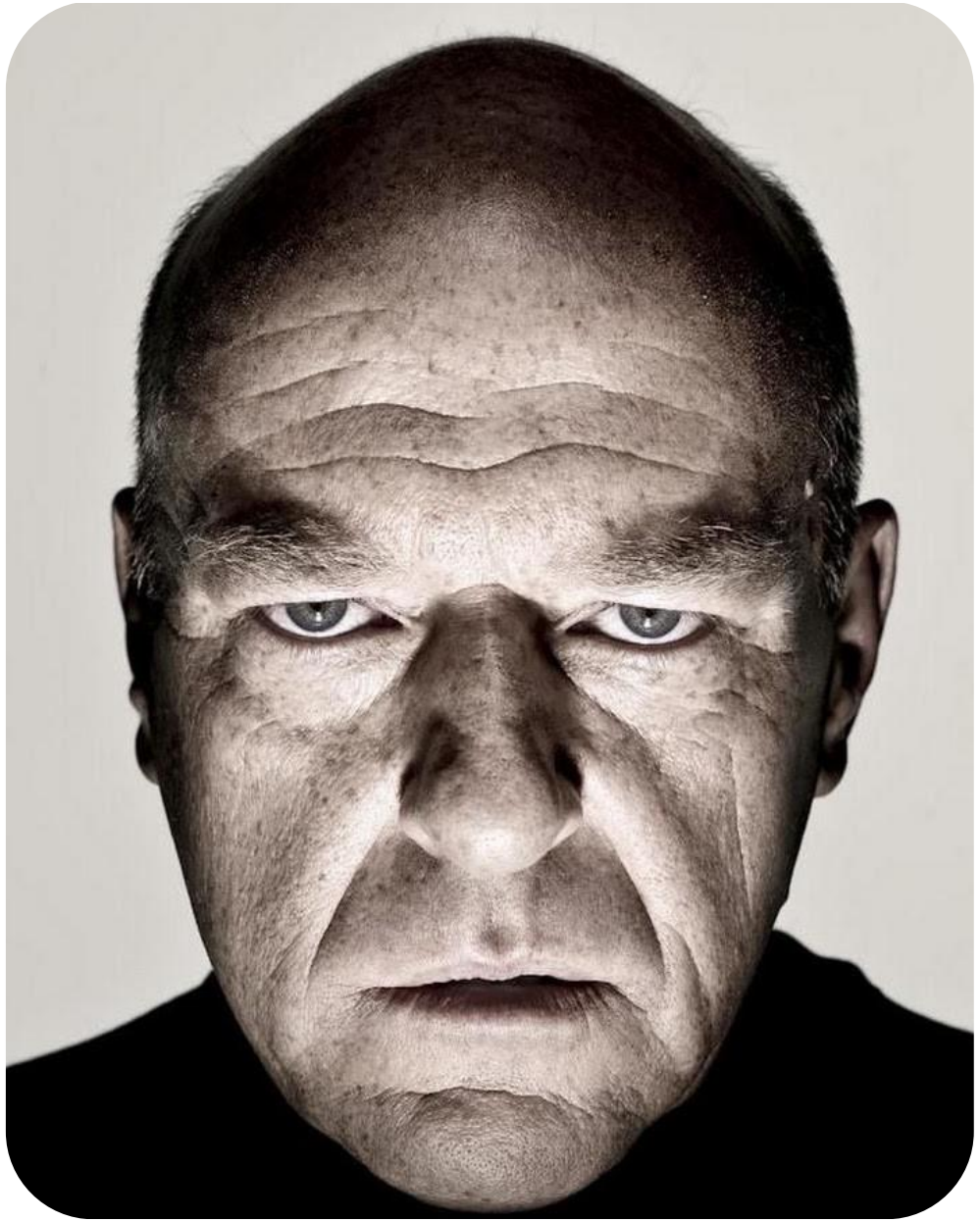
Источник изображения: адаптированный интернет-мем

«Архитектура-зомби»

Симптомы заражения проекта зомби-архитектурой

Вы часто слышите фразы:

- «Диаграмма не актуальна,
  но там в целом всё так..»
- «Лучше спросить у разрабов,
  в доке там не совсем так...»
- «Ну так исторически сложилось...»



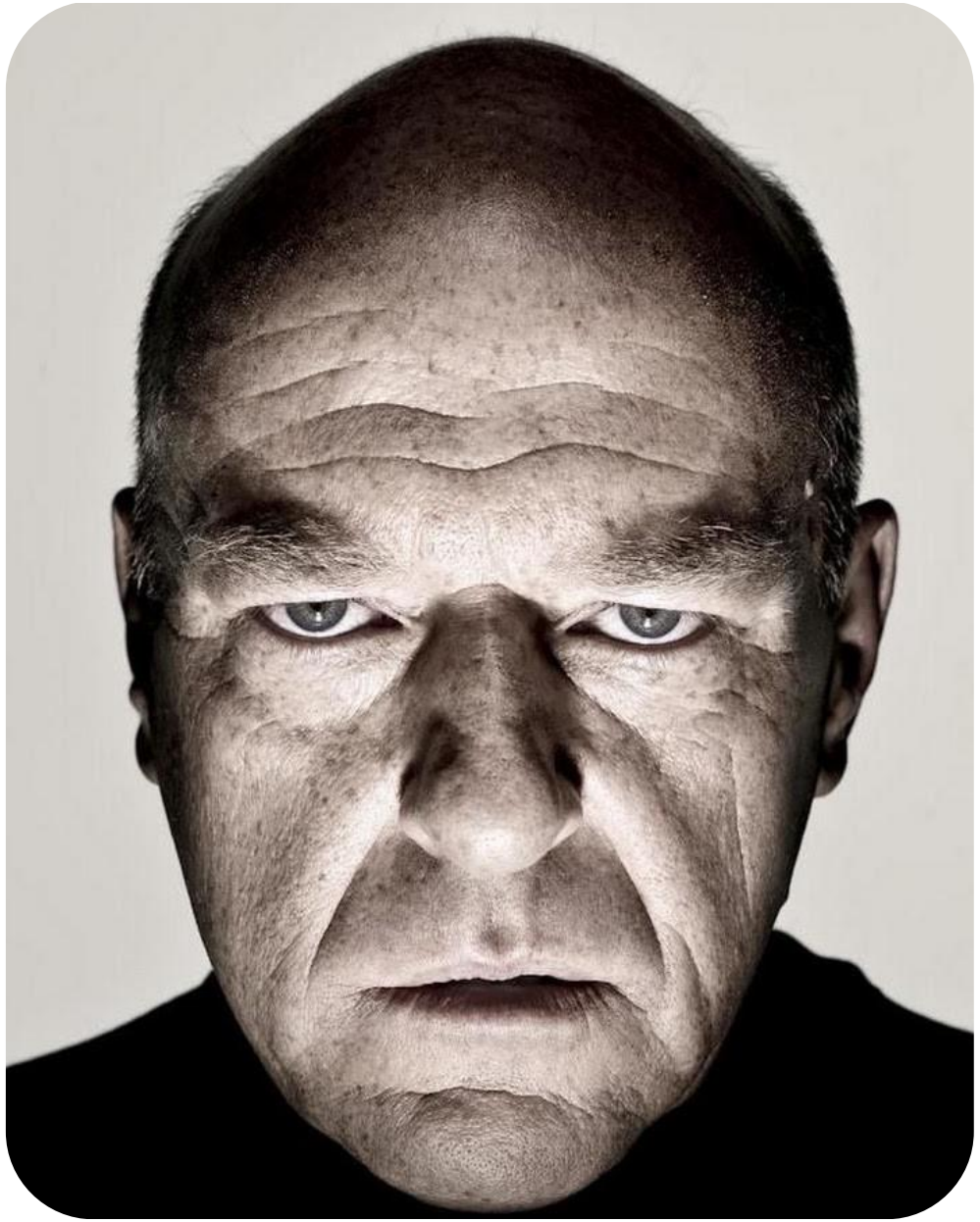
Источник изображения: адаптированный интернет-мем

«Архитектура-зомби»

Симптомы заражения проекта зомби-архитектурой

Вы часто слышите фразы:

- «Диаграмма не актуальна,
  но там в целом всё так..»
- «Лучше спросить у разрабов,
  в доке там не совсем так...»
- «Ну так исторически сложилось...»
- «Сюда можешь совсем не смотреть,
  тут всё не так...»

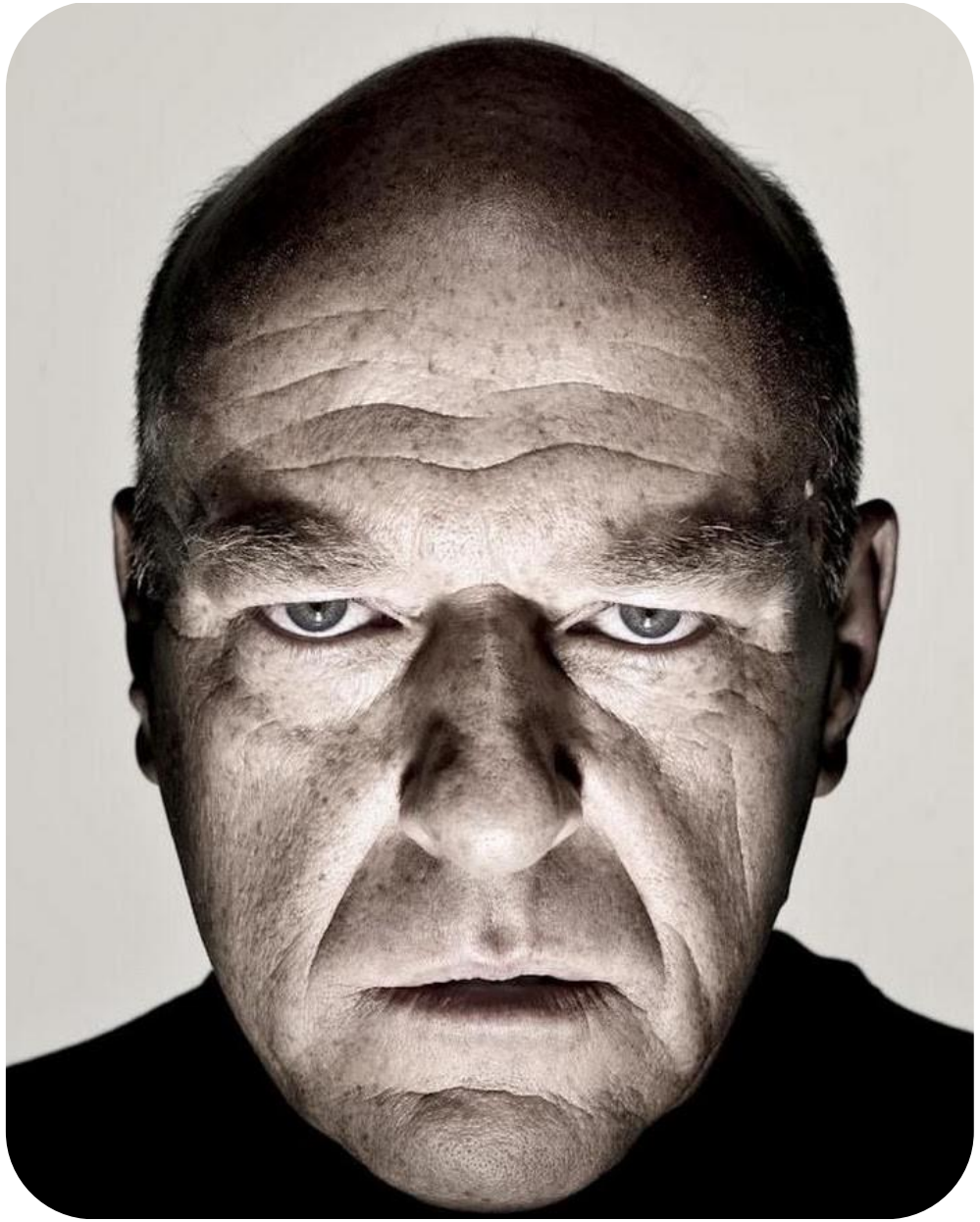


Источник изображения: адаптированный интернет-мем

«Архитектура-зомби»

Почему зомби так живучи?

- Нет процесса: обновление архитектуры — это не часть процесса, а подвиг

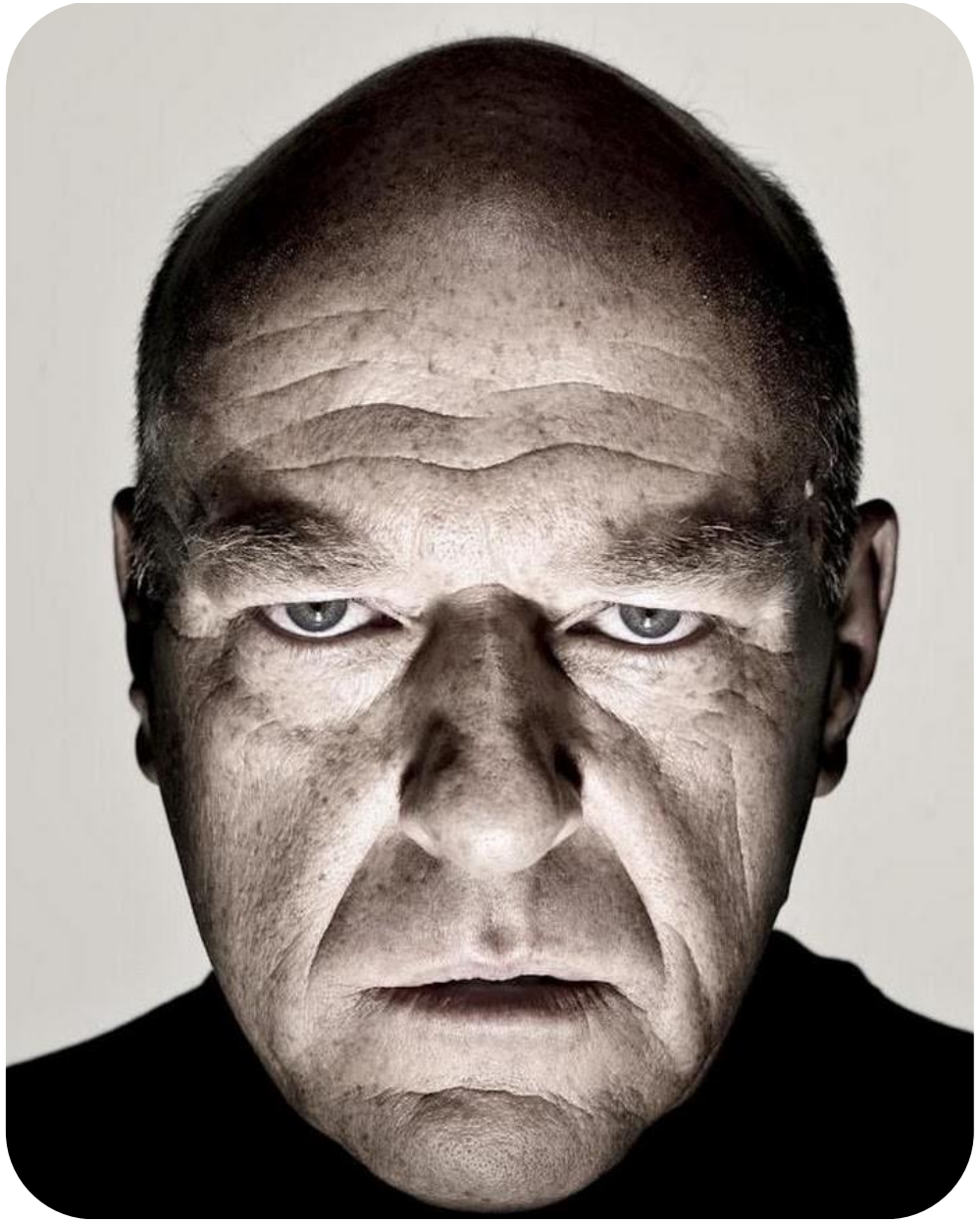


Источник изображения: адаптированный интернет-мем

«Архитектура-зомби»

Почему зомби так живучи?

- Нет процесса: обновление архитектуры — это не часть процесса, а подвиг
- «Одинокий архитектор»: знания о системе ушли вместе с ним в другой проект

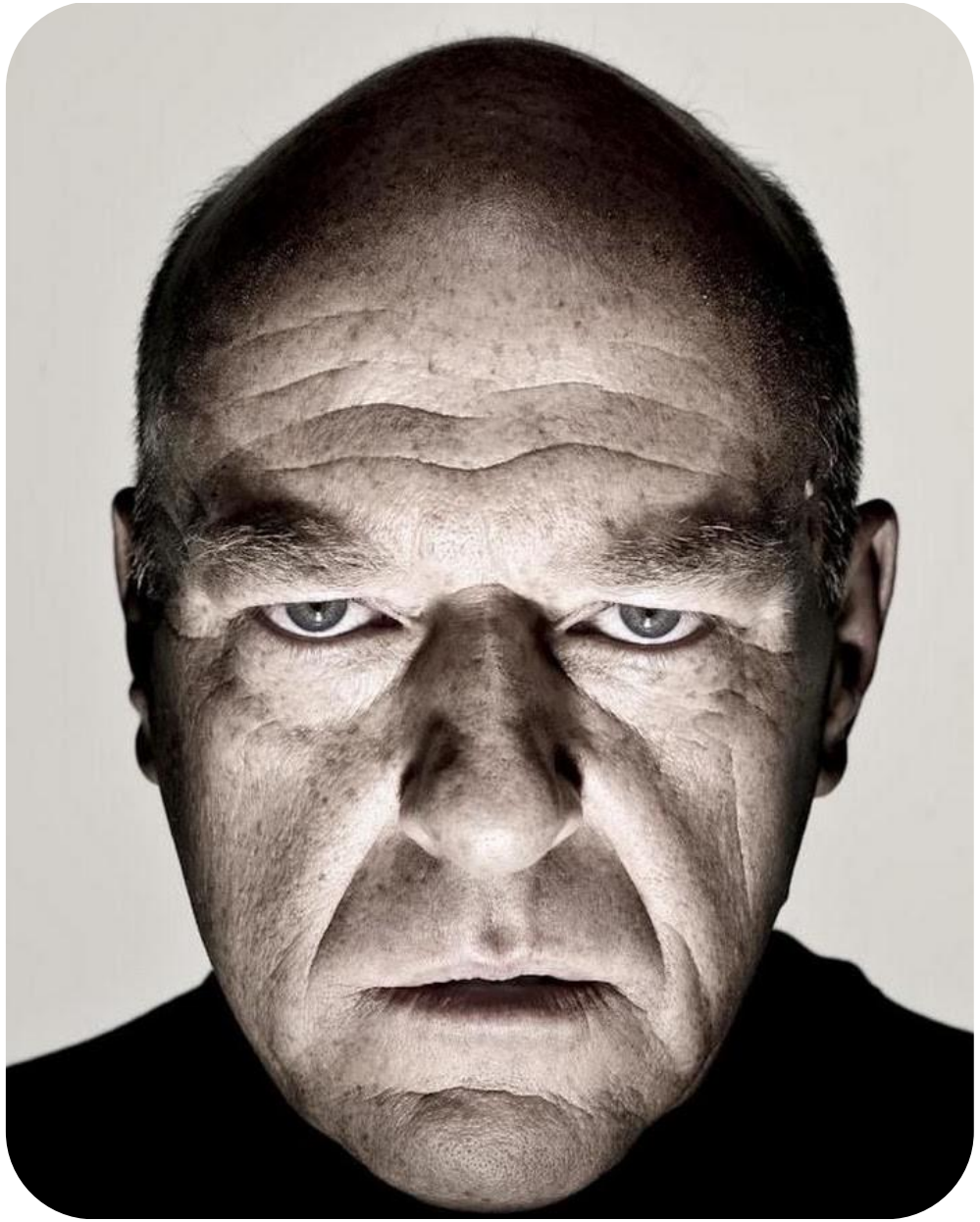


Источник изображения: адаптированный интернет-мем

«Архитектура-зомби»

Почему зомби так живучи?

- Нет процесса: обновление архитектуры — это не часть процесса, а подвиг
- «Одинокий архитектор»: знания о системе ушли вместе с ним в другой проект
- Всем плевать: команда не верит, что документация может быть полезной



Источник изображения: адаптированный интернет-мем

Еще одна история

Источник изображения: адаптированный интернет-мем

Коллективная ответственность

Архитектурный комитет:

Не суд, а рабочий инструмент

Коллективная ответственность

Архитектурный комитет:

Не суд, а рабочий инструмент

Регулярные встречи (раз в спринт)

Коллективная ответственность

Архитектурный комитет:

Не суд, а рабочий инструмент

Регулярные встречи (раз в спринт)

Цель — обсудить открытые MR с архитектурными изменениями

Коллективная ответственность

Архитектурный комитет:

Не суд, а рабочий инструмент

Регулярные встречи (раз в спринт)

Цель — обсудить открытые MR с архитектурными изменениями

Принимают участие:

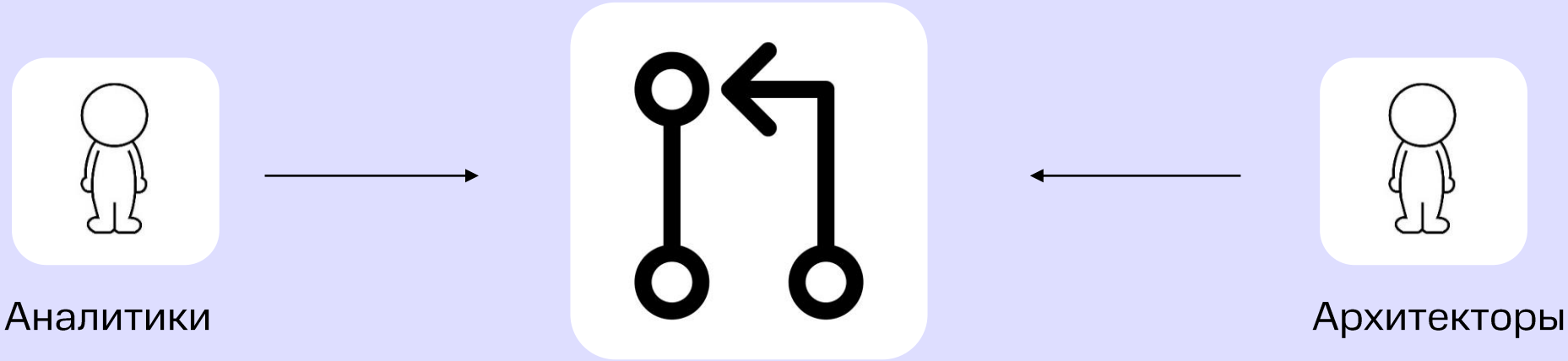
Аналитики

Архитекторы

Технические эксперты

Технические лидеры

Коллективная ответственность

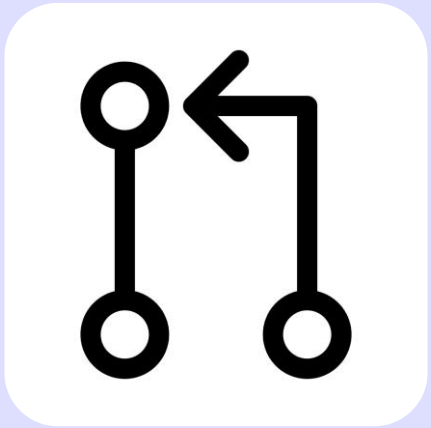


- Обсудить открытые MR
  с архитектурными изменениями
- Принять решение
  (одобрить/отклонить MR)
- Зафиксировать решение в ADR

Источник изображения: commons.wikimedia.org по лицензии MIT

Коллективная ответственность

Аналитики



Архитекторы

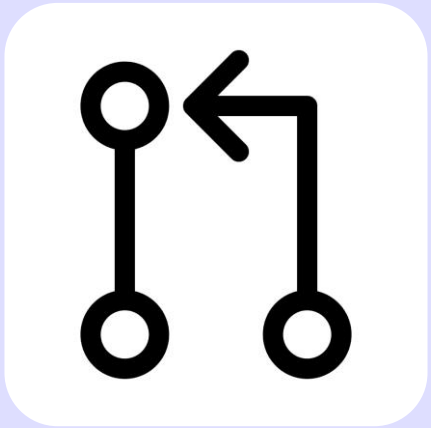
Технические лидеры

- Обсудить открытые MR с архитектурными изменениями
- Принять решение (одобрить/отклонить MR)
- Зафиксировать решение в ADR

Источник изображения: commons.wikimedia.org по лицензии MIT

Коллективная ответственность

Аналитики



Архитекторы

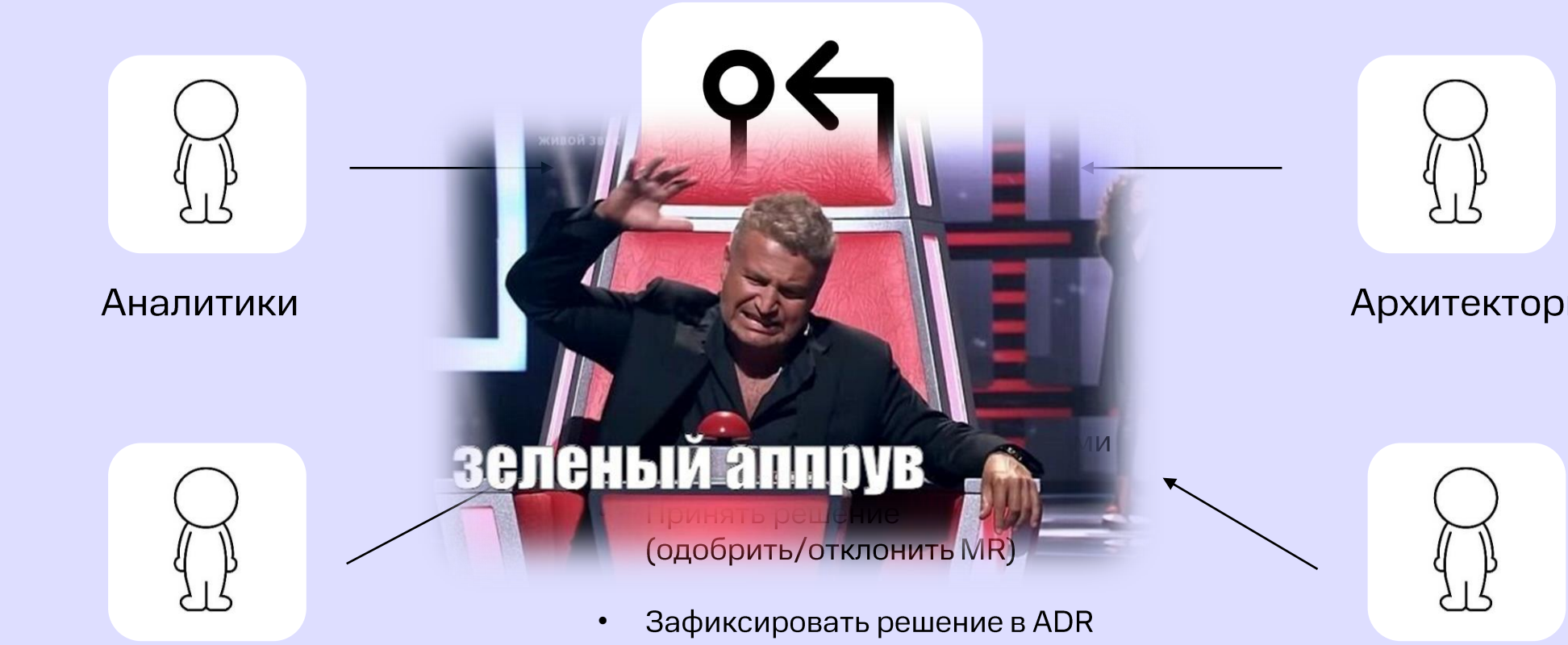
- Обсудить открытые MR с архитектурными изменениями
- Принять решение (одобрить/отклонить MR)
- Зафиксировать решение в ADR

Технические лидеры

Эксперты

Источник изображения: commons.wikimedia.org по лицензии MIT

Коллективная ответственность



- Принять решение (одобрить/отклонить MR)
- Зафиксировать решение в ADR

Источник изображения: адаптированный интернет-мем, commons.wikimedia.org по лицензии MIT

Коллективная ответственность

Техдолг?
Теперь только по предзаказу!

Раньше:

Технический долг возникал тихо и незаметно — как будто кто-то ночью принёс в код сомнительное решение и убежал

Техдолг?
Теперь только по предзаказу!

Раньше:

Технический долг возникал тихо и незаметно — как будто кто-то ночью принёс в код сомнительное решение и убежал

Теперь:

Чтобы создать техдолг, нужно пройти квест:

1. Описать
2. Защитить
3. Согласовать

Техдолг теперь — осознанный выбор, а не случайная ошибка.
Его всегда видно в ADR, и за него несёт ответственность вся команда!

Забудьте про «архитектуру-зомби»

Рисуйте кодом, а не вручную

C4 + DSL + Structurizr —
ваш ключ к актуальным и понятным
диаграммам без боли

Забудьте про «архитектуру-зомби»

Рисуйте кодом, а не вручную

C4 + DSL + Structurizr —
ваш ключ к актуальным и понятным диаграммам без боли

Храните всё в Git

ADR и код архитектуры должны жить в одном месте с вашим приложением

Забудьте про «архитектуру-зомби»

Рисуйте кодом, а не вручную

C4 + DSL + Structurizr —
ваш ключ к актуальным и понятным диаграммам без боли

Храните всё в Git

ADR и код архитектуры должны жить в одном месте с вашим приложением

Проводите review решений

Merge Request — это мощнейший инструмент для архитектурного диалога всей команды

Забудьте про «архитектуру-зомби»

Рисуйте кодом, а не вручную

C4 + DSL + Structurizr —
ваш ключ к актуальным и понятным диаграммам без боли

Храните всё в Git

ADR и код архитектуры должны жить в одном месте с вашим приложением

Проводите review решений

Merge Request — это мощнейший инструмент для архитектурного диалога всей команды

Договаривайтесь регулярно

Архитектурный комитет —
это не страшно, а полезно

Забудьте про «архитектуру-зомби»

- Больше не нужно гадать «как тут всё устроено?»
- Скорость онбординга новых людей взлетает
- Решения становятся общими и осознанными

Сделайте архитектуру живой — освободите время для решения реальных задач!

Почитать по теме

- **C4 Model**
  https://c4model.com — официальный сайт
  с объяснением нотации и примерами

- **Structurizr**
  https://structurizr.com — инструмент для «архитектуры как кода».
  Документация, примеры DSL

- **ADR GitHub Repository**
  https://github.com/joelparkerhenderson/architecture-decision-record —
  огромная коллекция материалов по ADR

- **C4**
  https://www.infoq.com/articles/C4-architecture-model—
  практическое руководство по внедрению

Спасибо за внимание!

Германов Павел

Руководитель направления по архитектуре
и системному анализу в МТС Финтех

Связаться со мной