DDD в действии

Максим Морев

Gazprombank

@maxology

@codemonsterslogs

JPoint

DDD в действии

Руководство с примерами на Kotlin по внедрению Предметно Ориентированного Проектирования (Domain Driven Design) в команду и обращения её в безумную машину по доставке чистого кода

Обсудим сложности

- не говорить про DDD
- внедрение в существующей команде
- рефакторинг существующего "старого кода" по кукбуку

DDD code toolkit: https://t.ly/pwgy

Унесешь с собой

- паттерн приложения с тестами на бизнес-логике и кучу полезных рекомендаций, которые работают

DDD:
самое важное

«The amazing thing about DDD was not the patterns or the practices, it was the quiet way it put the lie to a fundamental tenets of software engineering: the lie that programmers did not need to have an understanding of domains, that everything they needed to know was a set of requirements that the code must satisfy.»

Prof. David West

Код
побочный эффект коммуникации

Качественная разработка
результат качественной коммуникации,
а не постановки

Разработчик
эксперт предметной области

Прозрачность процессов
vs Job Safety

Свобода ?
уволиться одним днем

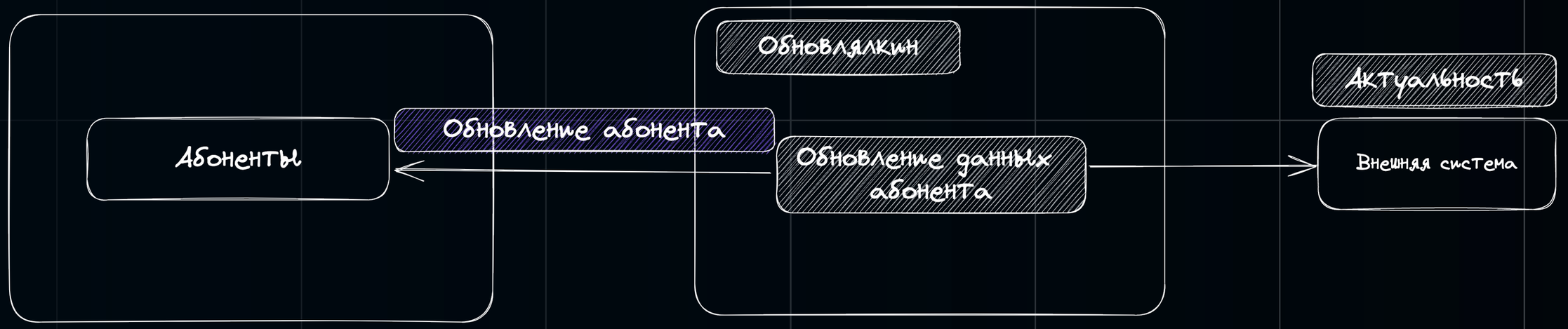
Стратегические паттерны

От контекста команды к задаче

Bounded Context

Ограничиваем контекст и учимся

находить место бизнес-логике



Ubiquity Language

Общий язык — единый язык в документации, коде,

в общении с экспертами домена

Jira

Создание абонента

Привязка лояльности

Обновление абонента

Организация процессов и документов

- Документация к сервисам – в исходниках
- Сопроводительную документацию собираем в одном разделе по бизнес-процессу

Сопроводительная Документация

- обновление данных
  - Описание процесса
  - Взаимодействия со смежными командами
  - Релиз
    - ИФТ
    - Дорожная карта ПСИ

Перенесем документацию в код

P.O.P Pipeline Oriented Programming
video

или

Data Flow Programming
wikipedia

Бенефиты

- код есть документация – просто читать не только программисту
- это всегда однонаправленный поток, даже с ответвлениями
- упрощает компоновку шагов
- помогает следовать принципам хорошего дизайна

Опиши верхнеуровнево бизнес-процесс в функциональном стиле

Описание должно просто ответить на вопрос:
Что происходит в системе по бизнес-процессу?

Пример:

Чтобы обновить данные абонента, необходимо:
| получить данные для обновления абонента
| запросить текущие данные абонента в системе
| сформировать запрос на обновление абонента
| отправить запрос обновления данных абонента

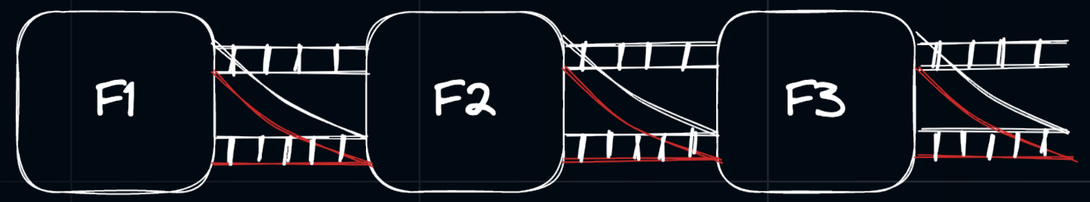
```
@Service
fun dataUpdateProcess(unvalidatedUpdateRequest:
UnvalidatedDataUpdateRequest)
: Mono<Result<SubscriberDataUpdateResponse>> =

    Mono.just(ValidatedDataUpdateRequest.emerge(unvalidatedUpdateRe
quest))
    .flatMap { findDataWithUpdates(it) }
    .flatMap { findSubscriberForUpdate(it) }
    .flatMap { prepareSubscriberUpdateRequest(it) }
    .flatMap { updateSubscriber(it) }


// > Result IN > Result OUT
private fun prepareSubscriberUpdateRequest(
    subscriberDataUpdate: Result<SubscriberDataUpdate>
): Mono<Result<SubscriberUpdateRequest>> =
subscriberDataUpdate.fold(
    onSuccess = { it.prepareUpdateRequest() },
                    //^ Бизнес-логика в Доменном классе
    onFailure = { Result.failure(it) }
                    //^ ошибку пробрасываем далее по пайпу процесса
  ).toMono()
```

DDD в функциональном стиле собираем благодаря паттернам

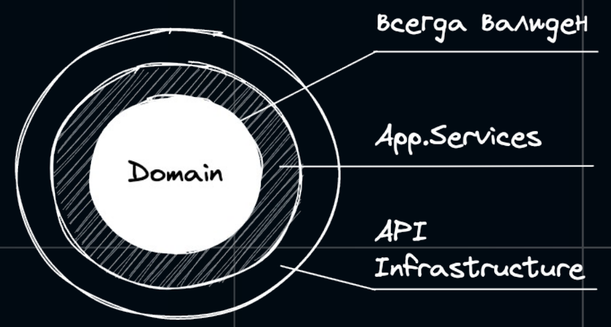
R.O.P Railway Oriented Programming in Error Handling



Сильная Доменная модель | Rich Domain Model

Type Driven Development

Onion Architecture



TDD: Классическая школа Тестирования и совсем немного лондонского вайба

Тактические паттерны DDD: самое важное

Строим благодаря

Type Driven Development

тактические паттерны

ValueObject

основной строительный блок модели

Aggregate

там где рождается изменение

Рецепт

- Стань экспертом предметной области — разберись, что и как должно работать на всех уровнях. Твой код — твоя ответственность.
- Опиши в функциональном стиле бизнес-процесс с доменными классами
- Реализуй в функциональном стиле всегда валидную Богатую Доменную Модель без примитивов
- Покрой юнит-тестами бизнес-логику, которая содержится в Доменной Модели
- Запусти Доменную Модель по тоннелю «бизнес-процесс» без исключений, на шлюзах поможет two track type Result<Data, Error> и canExecute/execute

И ПОМНИ...

Качественная разработка — это результат качественной коммуникации!

Максим Морев

@maxology

Полная версия доклада

Точка входа в мои эссе
https://t.me/codemonsterslogs

Bio

- Software Craftsamnship энтузиаст
- DDD, TDD Advocate
- CTO Стрима Госпроекты (Цифровой Рубль) в Газпромбанке