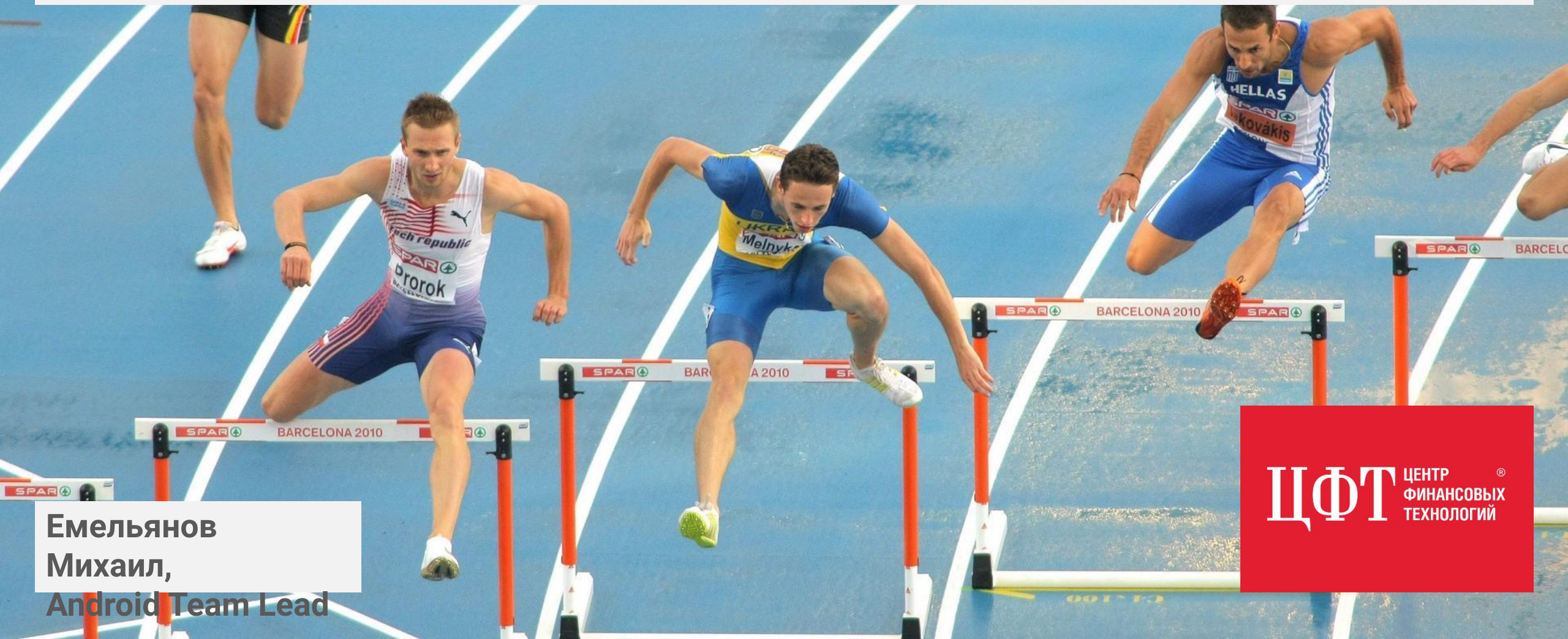


# Как переписать приложение с нуля и не потерпеть фиаско



Емельянов  
Михаил,  
Android Team Lead

ЦФТ  
ЦЕНТР  
ФИНАНСОВЫХ  
ТЕХНОЛОГИЙ



# О себе

7 лет в Android-разработке

В ЦФТ с 2013 г.

Разработал более десятка проектов



# О чем буду говорить

1. С чего все начиналось
2. Как набирали команду и решали задачи
3. Оцениваем масштаб разработки
4. Проблемы с архитектурой и технологиями
5. Какая польза от Unit-тестов и других инструментов
6. Не успевали к срокам, что делать

# О чем буду говорить

1. С чего все начиналось
2. Как набирали команду и решали задачи
3. Оцениваем масштаб разработки
4. Проблемы с архитектурой и технологиями
5. Какая польза от Unit-тестов и других инструментов
6. Не успевали к срокам, что делать



# О чем буду говорить

1. С чего все начиналось
2. Как набирали команду и решали задачи
- 3. Оцениваем масштаб разработки**
4. Проблемы с архитектурой и технологиями
5. Какая польза от Unit-тестов и других инструментов
6. Не успевали к срокам, что делать

# О чем буду говорить

1. С чего все начиналось
2. Как набирали команду и решали задачи
3. Оцениваем масштаб разработки
- 4. Проблемы с архитектурой и технологиями**
5. Какая польза от Unit-тестов и других инструментов
6. Не успевали к срокам, что делать

# О чем буду говорить

1. С чего все начиналось
2. Как набирали команду и решали задачи
3. Оцениваем масштаб разработки
4. Проблемы с архитектурой и технологиями
- 5. Какая польза от Unit-тестов и других инструментов**
6. Не успевали к срокам, что делать

# О чем буду говорить

1. С чего все начиналось
2. Как набирали команду и решали задачи
3. Оцениваем масштаб разработки
4. Проблемы с архитектурой и технологиями
5. Какая польза от Unit-тестов и других инструментов
6. Не успевали к срокам, что делать



**С чего все началось?**

# Бизнес поставил задачу

- Масштабироваться
- Увеличить скорость разработки фич
- Новый дизайн
- Улучшить стабильность

# Бизнес поставил задачу

- Масштабироваться
- Увеличить скорость разработки фич
- Новый дизайн
- Улучшить стабильность

# Бизнес поставил задачу

- Масштабироваться
- Увеличить скорость разработки фич
- **Новый дизайн**
- Улучшить стабильность

# Бизнес поставил задачу

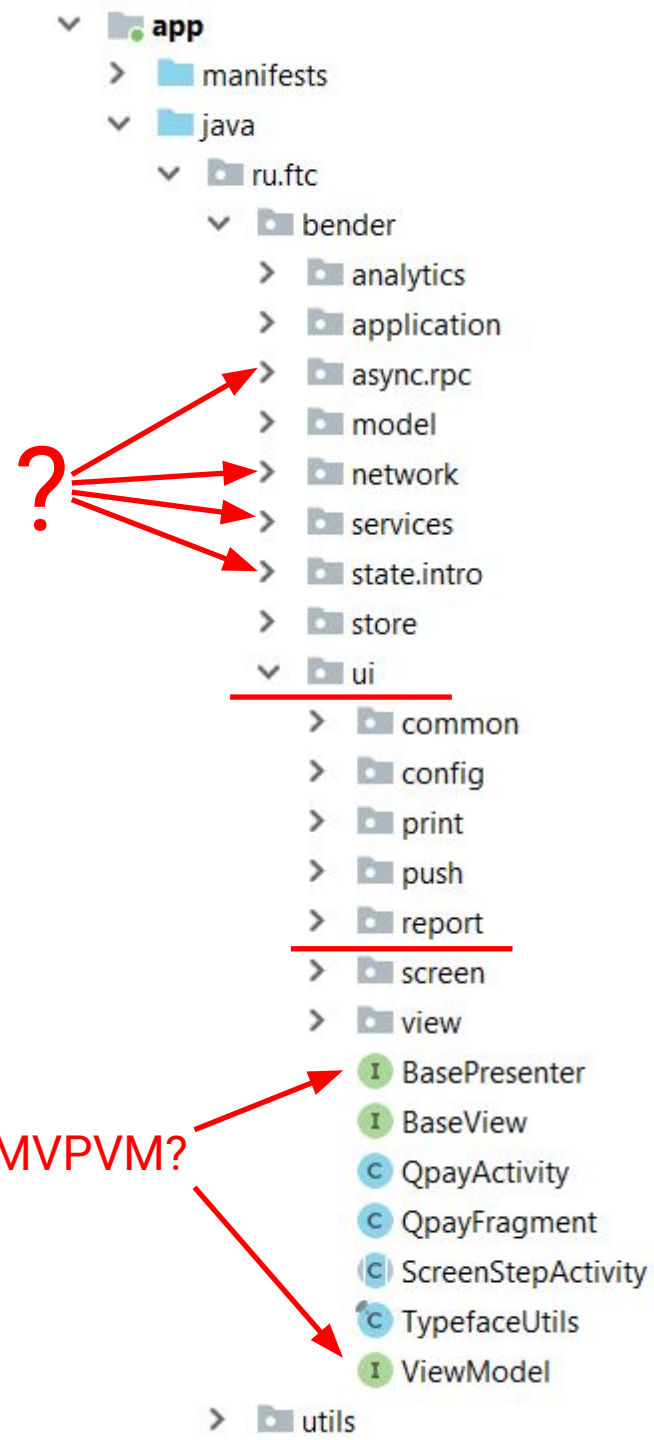
- Масштабироваться
- Увеличить скорость разработки фич
- Новый дизайн
- Улучшить стабильность

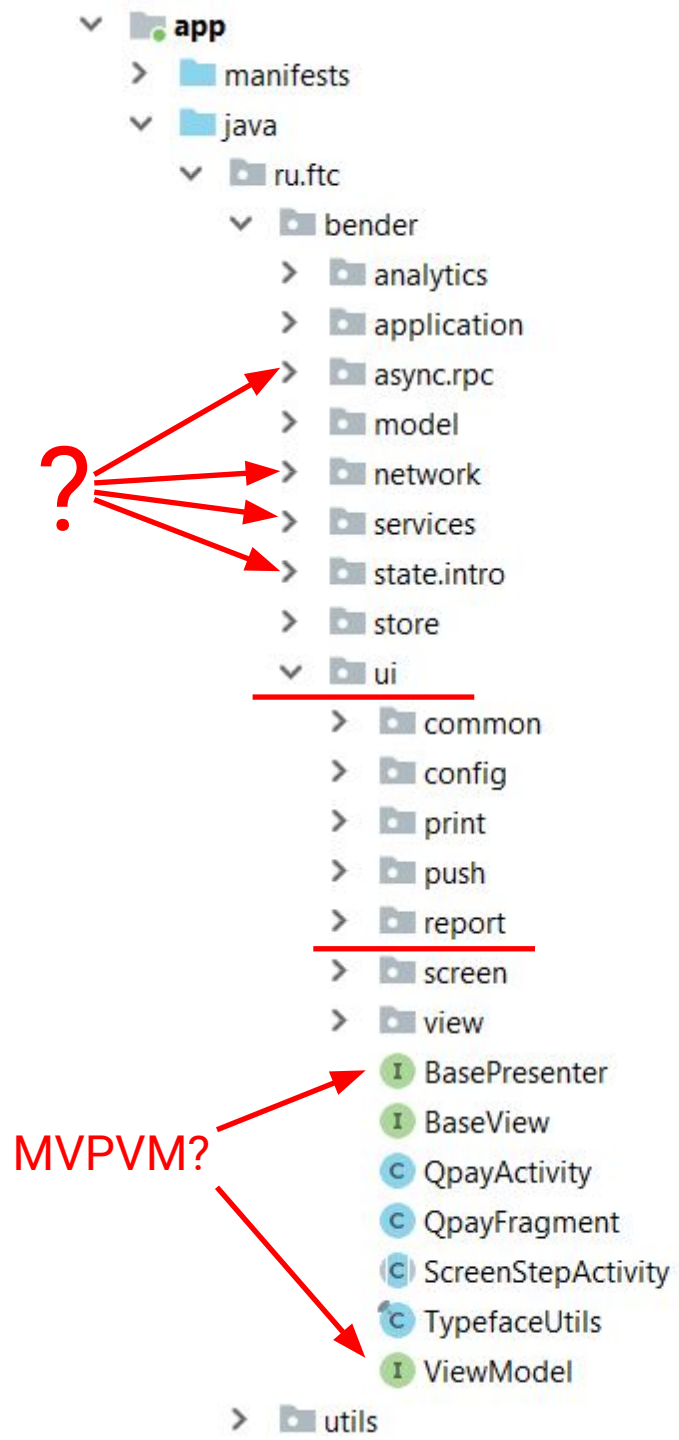
**Но не все так просто...**

# Проблема №1

**Архитектура**







```
class FragmentPaymentViewModel extends FragmentPaymentViewModelPartial implements QpayWebViewClient.OnLoading
```

```
private UrlsWrapper urls;
```

```
private final QpayWebViewClient webViewClient;
```

```
private OnPaymentActionSuccessListener onPaymentActionSuccessListener;
```

```
private final BaseFragment fragment;
```

```
FragmentPaymentViewModel(final Bundle state, final BaseFragment fragment, final UrlsWrapper urls) {
```

```
    super(state, fragment.getActivity());
```

```
    this.fragment = fragment;
```



```
webView().getSettings().setSupportMultipleWindows(true);
webView().setWebChromeClient(onCreateWindow(webView, dialog, userGesture, resultMessage) → {
    final WebView.HitTestResult result = webView.getHitTestResult();
    final String data = result.getExtra();
    if (data != null) {
        final Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(data));
        getContext().startActivity(browserIntent);
    }
    return false;
});
```

Стартуем Activity

```
setWebClient(webViewClient);
```

Делаем операции View

```
getView().setFocusableInTouchMode(true);
```

```
getView().requestFocus();
```

```
getView().setOnKeyListener((view, keyCode, keyEvent) → {
```

```
    if (keyCode == KeyEvent.KEYCODE_BACK) {
```

```
        if (getContext() != null) {
```

```
            getContext().sendBroadcast(new Intent(ScreenTransferBaseActivity.ACTION_POP_BACK_STACK));
```

```
        }
```

```
        return true;
```

```
    } else {
```

```
        return false;
```

```
    }
```

```
});
```

Отправляем broadcast

# Проблема №2

**Ресурсы и стили**

- <> styles\_component\_common.xml
- <> styles\_component\_countries\_list.xml
- <> styles\_component\_country\_list.xml
- <> styles\_component\_country\_selector.xml
- <> styles\_component\_drawer\_header\_view.xml
- <> styles\_component\_identity\_card.xml
- <> styles\_component\_intro\_slide.xml
- <> styles\_component\_near\_place\_item.xml
- <> styles\_component\_screen\_contact\_item.xml
- <> styles\_component\_transfer\_info\_field.xml
- <> styles\_component\_transfer\_sum.xml
- <> styles\_components\_common.xml
- <> styles\_fragment\_change\_password.xml
- <> styles\_fragment\_cities\_list.xml
- <> styles\_fragment\_successful\_page.xml
- <> styles\_fragment\_transfer\_confirmation.xml
- <> styles\_fragment\_transfer\_info.xml
- <> styles\_fragment\_transfer\_recipient.xml
- <> styles\_fragment\_transfer\_status.xml
- <> styles\_history.xml
- <> styles\_item\_screen\_cities\_list.xml
- <> styles\_navigation\_drawer.xml
- <> styles\_print\_transfer\_info.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="CountrySelectorDialogTitle" parent="BoldTextView">
    <item name="android:textColor">@color/black_87</item>
    <item name="android:textSize">@dimen/text_size_20</item>
  </style>
</resources>
```



```
<color name="light_gray_DD">#FFDDDDDD</color>
<color name="light_gray_FA">#FFFAFAFA</color>
<color name="light_grey_F8">#FFF8F8F8</color>
<color name="light_grey_AC">#FFACACAC</color>
<color name="light_gray_EB">#FFEBEBEB</color>
<color name="red">#FFE0002E</color>
<color name="blue">#FF4A90E2</color>
<color name="black">#FF000000</color>
<color name="black_12">#1F000000</color>
<color name="black_30">#30000000</color>
<color name="black_87">#DD000000</color>
<color name="black_54">#89000000</color>
<color name="black_7001">#B2010101</color>
<color name="black_70">#B2000000</color>
<color name="black_97">#97000000</color>
<color name="black_3D">#FF3D3D3D</color>
<color name="black_1F">#1F000000</color>
<color name="black_0B">#FF0B0B0B</color>
<color name="black_DE">#DE272a32</color>
<color name="black_06">#FF060606</color>
<color name="black_DE00">#DE000000</color>
<color name="black_8A00">#8a000000</color>
<color name="accent_blue">#FF536DFE</color>
<color name="shadow_color">#63000000</color>
<color name="yellow">#FFFFFF66D</color>
<color name="yellowC732">#ffc732</color>
<color name="battleship_gray">#FF6A6f7B</color>
<color name="gray_F2">#FFF2F2F2</color>
<color name="gray_74">#FF747474</color>
<color name="gray_38">#FF383838</color>
```





# Проблема №3

## Рефакторинг

**Смотрим покрытие тестами**

ru.ftc 10% classes, 8% lines covered



**В результате получили**

# Монолитный проект – сложно масштабироваться

Нет архитектуры – кругом спагетти-код

Нет тестов – делаем одно, ломаем другое

Долгий рефакторинг – тонем в техдолге

Невозможен – «просто» редизайн

Монолитный проект – сложно масштабироваться

**Нет архитектуры – кругом спагетти-код**

Нет тестов – делаем одно, ломаем другое

Долгий рефакторинг – тонем в техдолге

Невозможен – «просто» редизайн

Монолитный проект – сложно масштабироваться

Нет архитектуры – кругом спагетти-код

**Нет тестов – делаем одно, ломаем другое**

Долгий рефакторинг – тонем в техдолге

Невозможен – «просто» редизайн



Монолитный проект – сложно масштабироваться

Нет архитектуры – кругом спагетти-код

Нет тестов – делаем одно, ломаем другое

**Долгий рефакторинг – тонем в техдолге**

Невозможен – «просто» редизайн

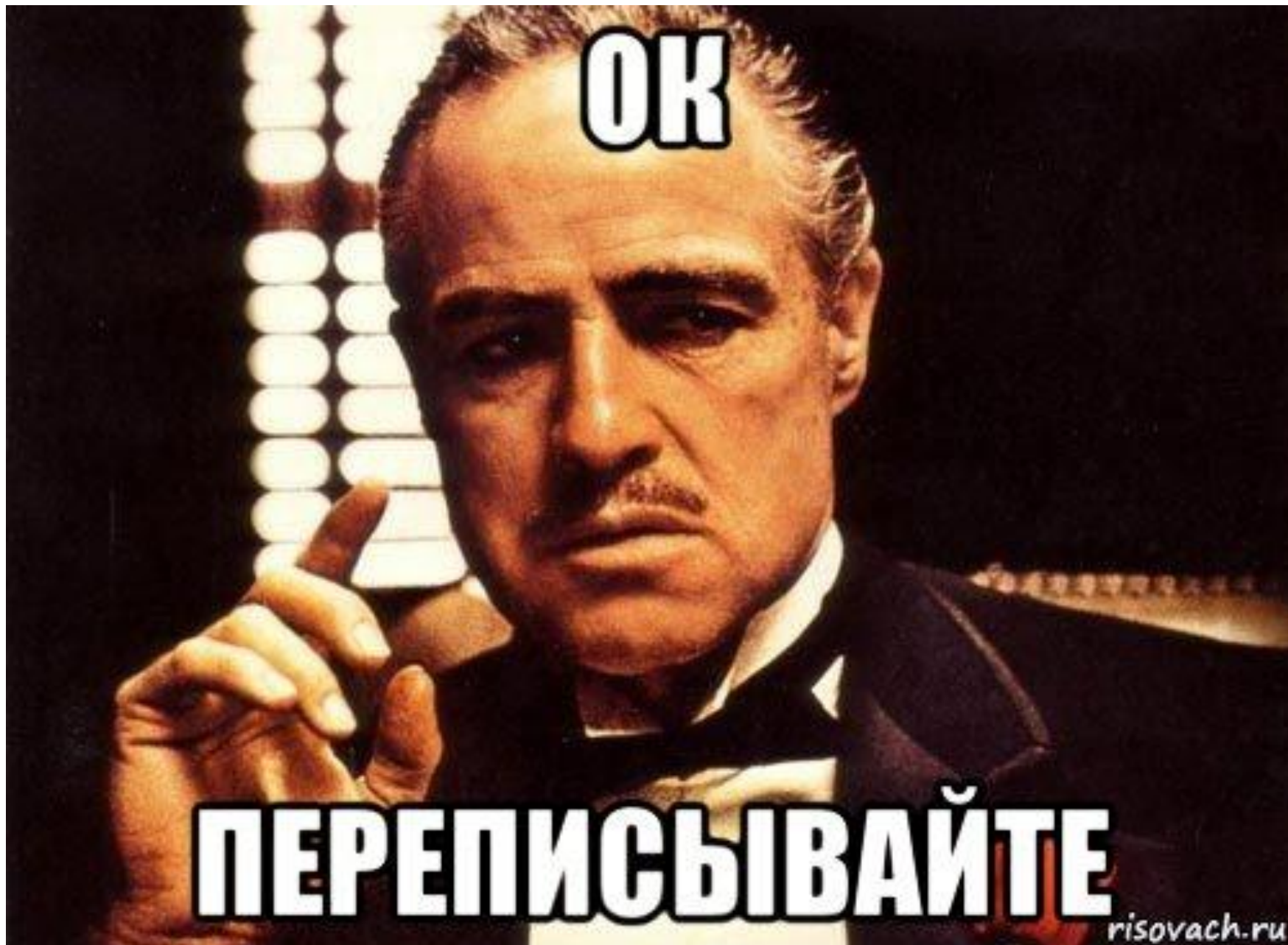
Монолитный проект – сложно масштабироваться

Нет архитектуры – кругом спагетти-код

Нет тестов – делаем одно, ломаем другое

Долгий рефакторинг – тонем в техдолге

**Невозможен – «просто» редизайн**



OK

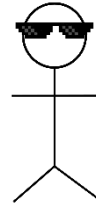
ПЕРЕПИСЫВАЙТЕ

risovach.ru

# Набираем команду, решаем задачи

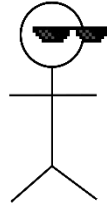


Вася – Senior



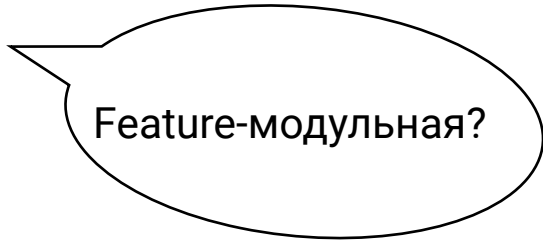
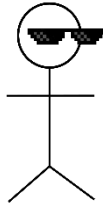
Модульная

Петя – Senior

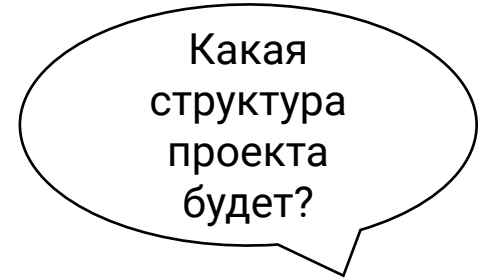


Монолит

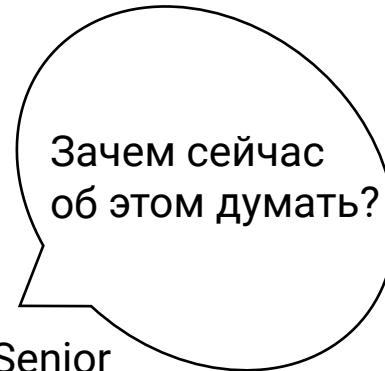
Марк – тоже Senior



Feature-модульная?

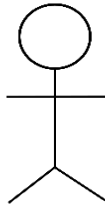


Какая  
структура  
проекта  
будет?



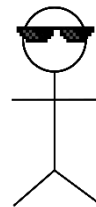
Зачем сейчас  
об этом думать?

Кирилл – Junior

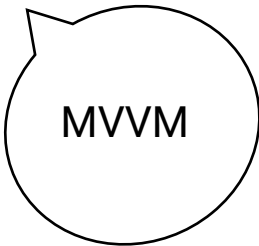
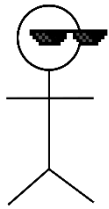


Как скажете

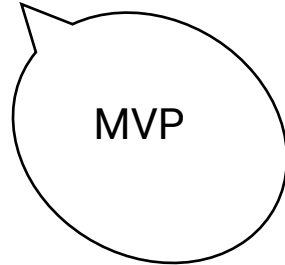
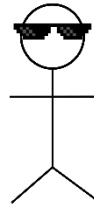
Иннокентий – почти Senior



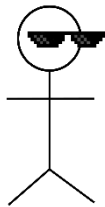
Петя – Senior



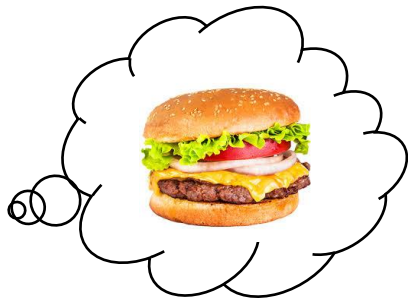
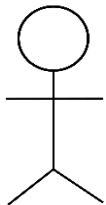
Вася – Senior



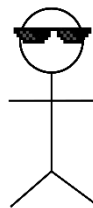
Марк – тоже Senior



Кирилл – Junior

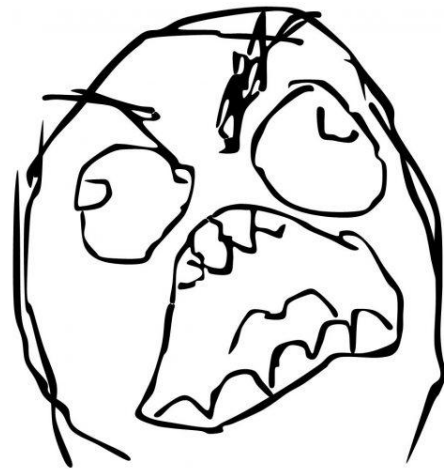


Иннокентий – почти Senior





**Прошло много часов,  
ни одного решения**



**Правило  
«Команда из сильных  
разработчиков,  
не всегда делает разработку  
сильной»**



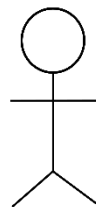
**Решение**

**Уволить всех Senior?**

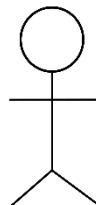
**Решение**

**Оставить 1-го**

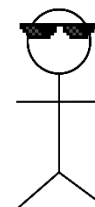
Ваня – Middle



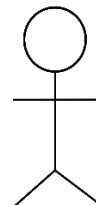
Джон – Middle



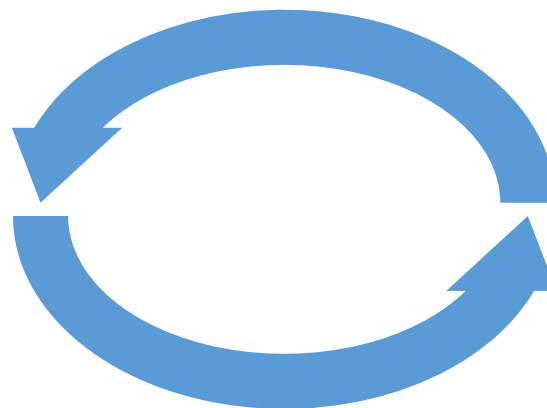
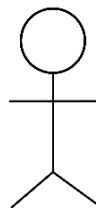
Вася – Senior



Кирилл – Junior



Гриша – Middle



# Критерии эффективной команды

# Единомышленники

1 tech lead

Уровень знаний не ниже  
среднего

Прагматичный выбор технологий

1 команда = 4-5  
человек

Единомышленники

**1 tech lead**

Уровень знаний не ниже  
среднего

Прагматичный выбор технологий

1 команда = 4-5  
человек

Единомышленники

1 tech lead

**Уровень знаний не ниже  
среднего**

Прагматичный выбор технологий

1 команда = 4-5  
человек

Единомышленники

1 tech lead

Уровень знаний не ниже  
среднего

**Прагматичный выбор технологий**

1 команда = 4-5  
человек



Единомышленники

1 tech lead

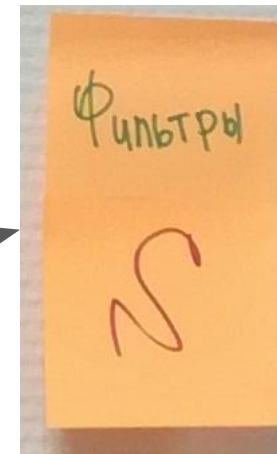
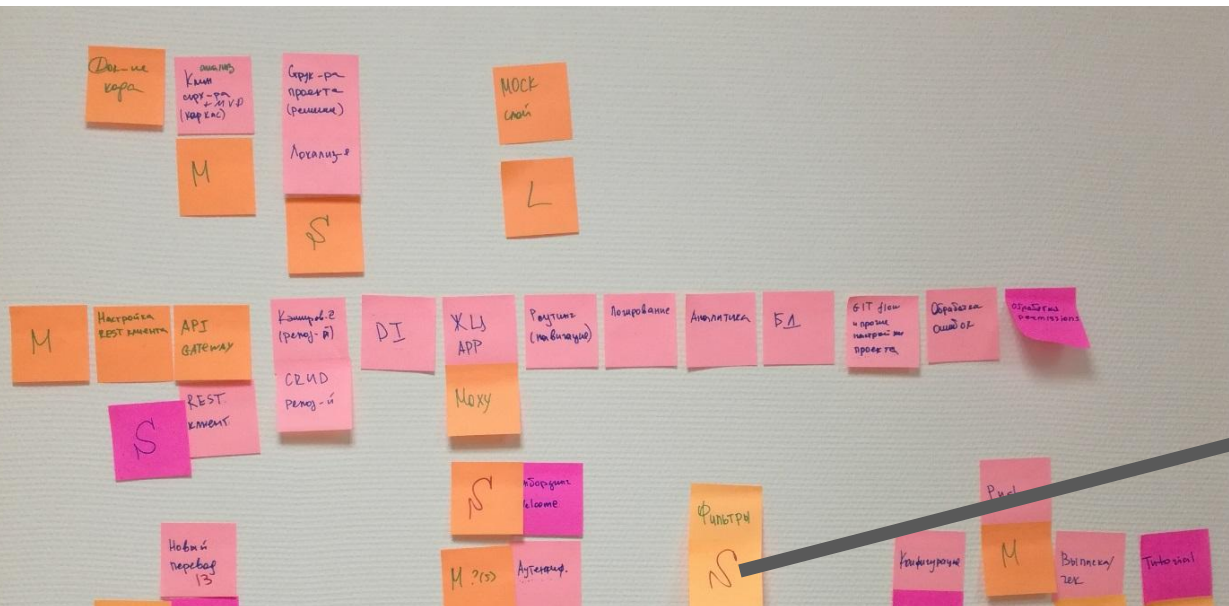
Уровень знаний не ниже  
среднего

Прагматичный выбор технологий

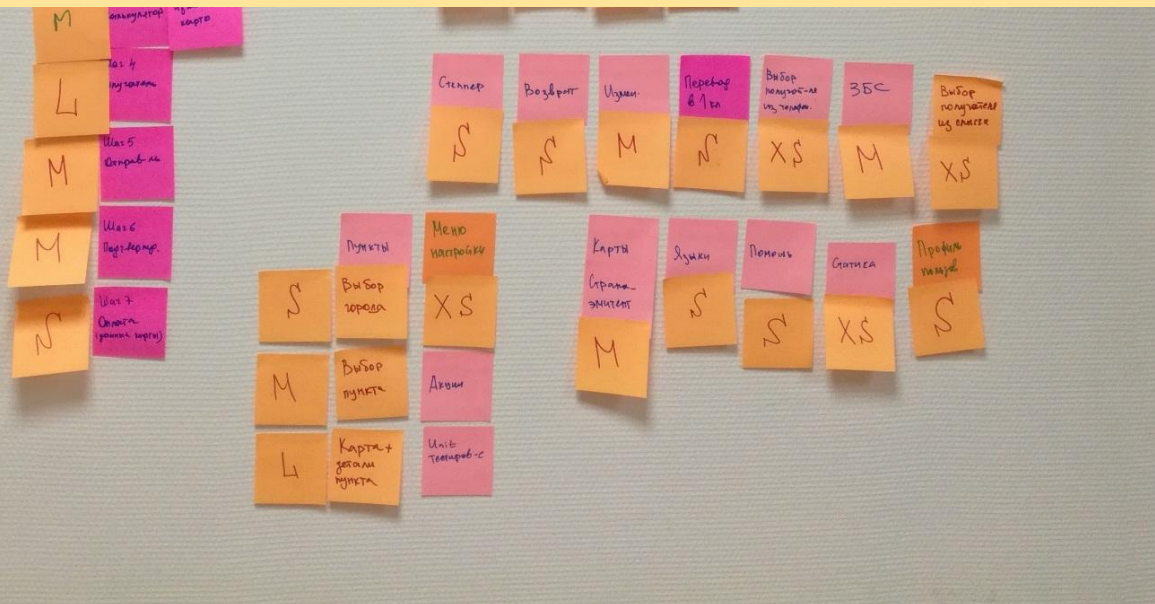
**1 команда = 4-5  
человек**

# Разбираем приложение по кирпичикам





~ 5 месяцев



- Оценка на 1 чел (S, M, L, XL), участвует вся команда

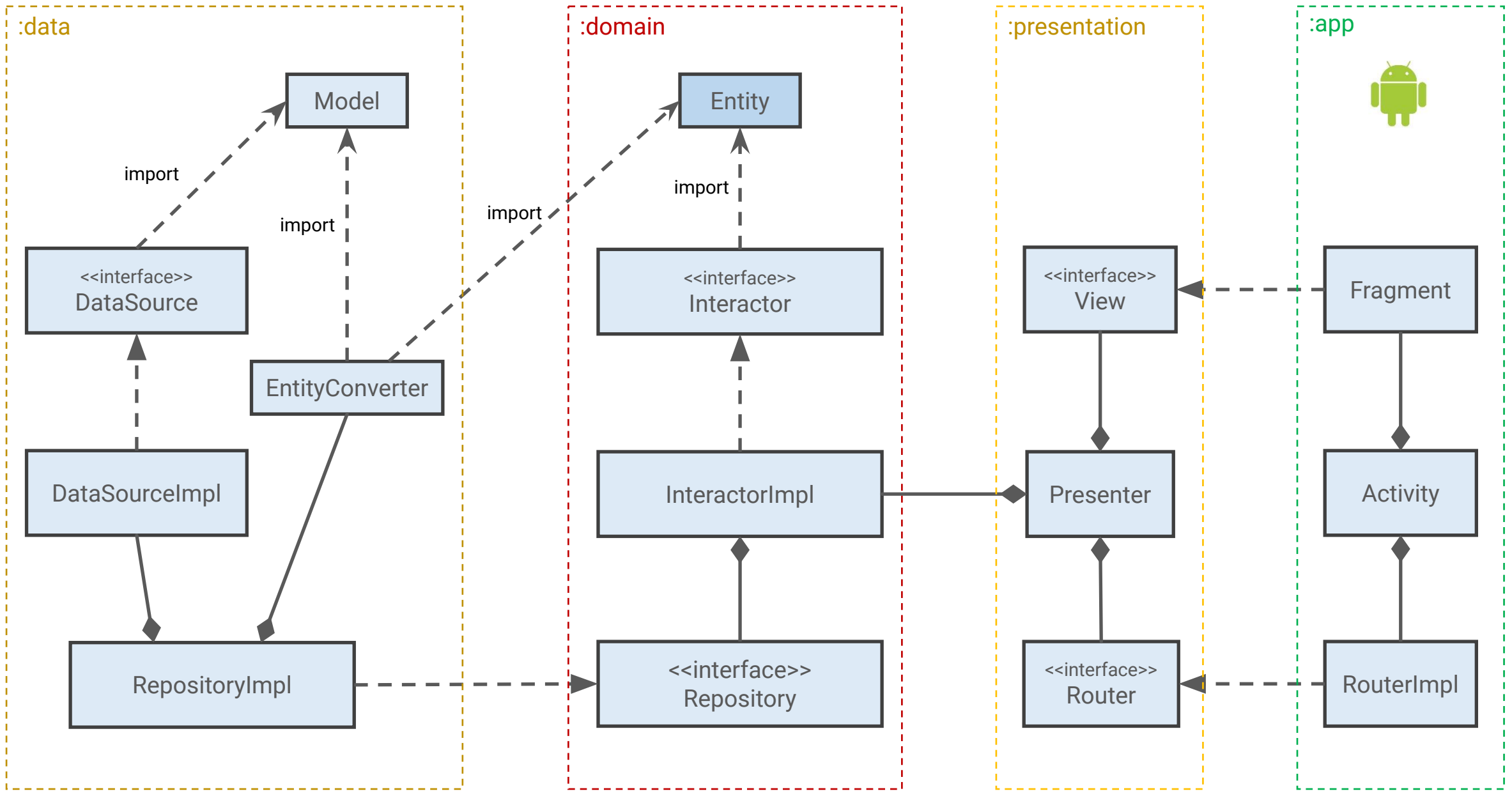
- Считаем, отдаем бизнесу

# Строим архитектуру



# Требования к архитектуре

- Масштабируется
- Бизнес-логика отделена от представления и данных
- Не зависит от реализации: UI, библиотек, платформы
- Тестируется
- Простая в понимании и применении





**Но мы наступили на грабли :(**



# Грабли №1

**Бизнес-логика == представлению**



```
interface LoginInteractor {
```

```
    fun requestOtp(phone: Phone): Completable
```

```
    fun getCountryByKnownPhone(): Single<Country>
```

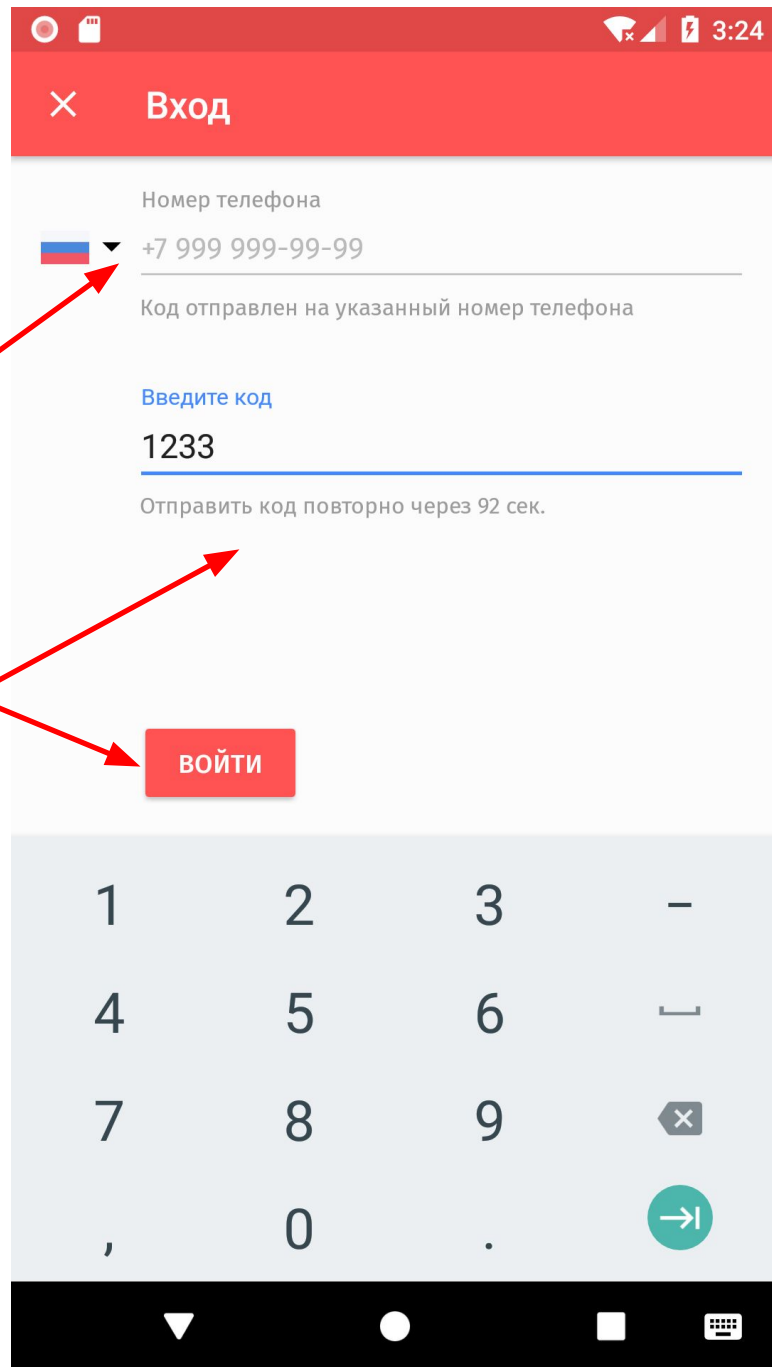
```
    fun createSession(phone: Phone, password: String): Completable
```

```
    fun timerFor(phone: Phone): Observable<Long>
```

```
    fun getKnownPhone(): Single<Phone>
```

```
    fun onCountrySelect(country: Country): Completable
```

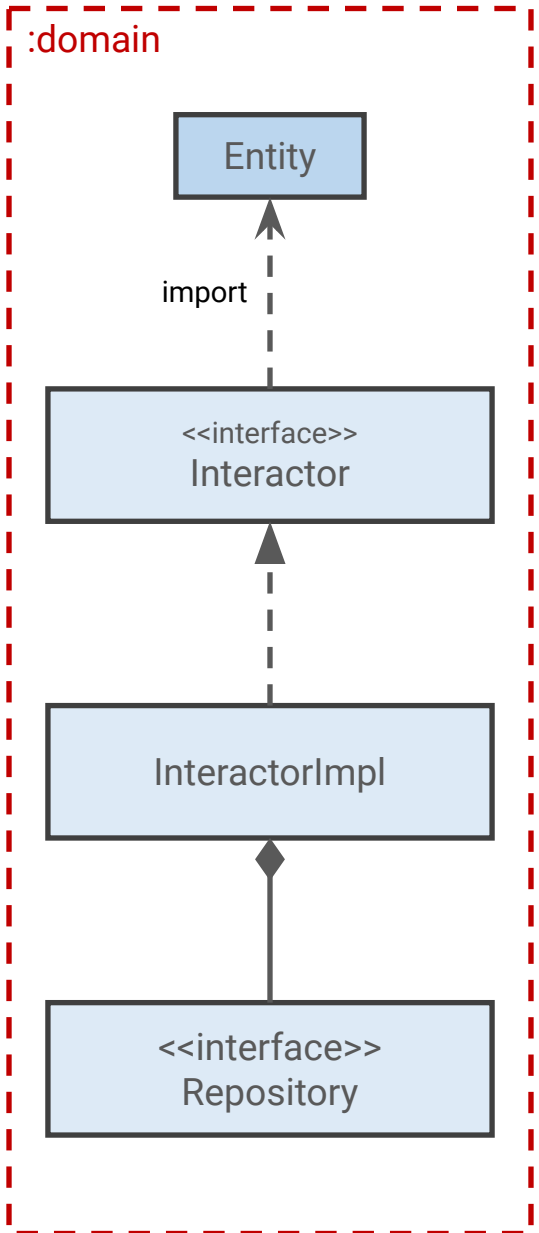
```
}
```

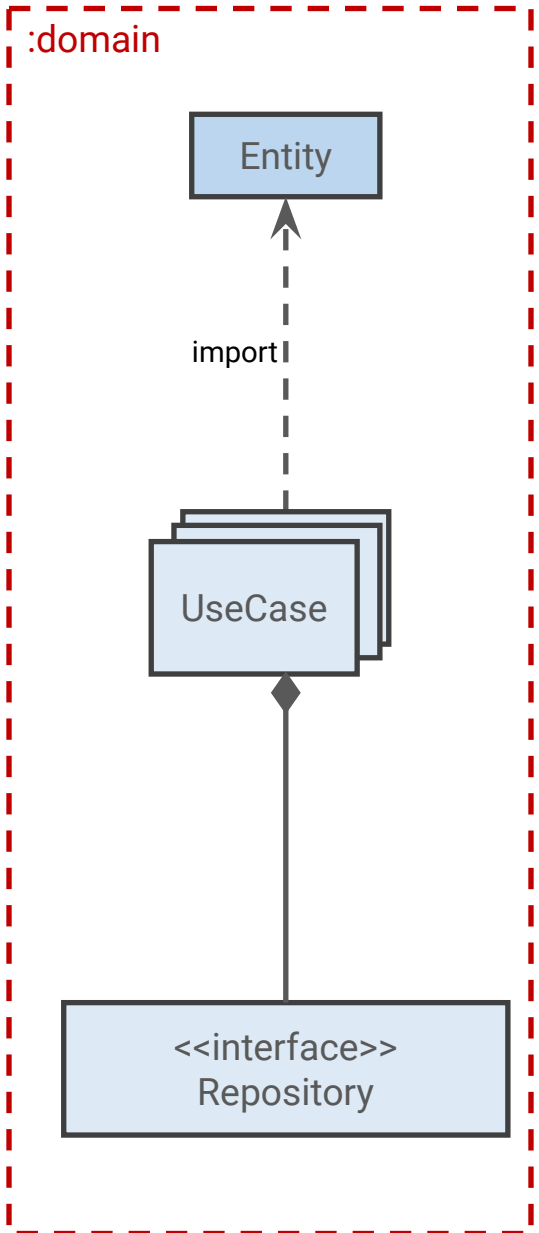


```
} class LoginInteractorImpl @Inject constructor(  
    private val pushInteractor: PushInteractor,  
    private val phoneComponentInteractor: PhoneComponentInteractor,  
    private val sessionRepository: SessionRepository,  
    private val appSettingsRepository: AppSettingsRepository,  
    private val phoneSettingsRepository: PhoneSettingsRepository,  
    private val validationInteractor: ValidationInteractor,  
    private val clock: Clock,  
    private val eda: Eda,  
    private val analytics: LoginAnalytics,  
    private val analyticsConfiguration: AnalyticsConfiguration  
) : LoginInteractor {
```

**Решение**

**Переделываем на UseCase**





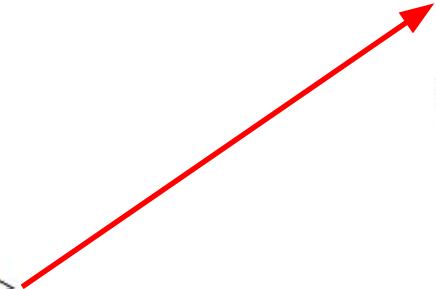
```
interface LoginInteractor {  
  
    fun requestOtp(phone: Phone): Completable  
  
    fun getCountryByKnownPhone(): Single<Country>  
  
    fun createSession(phone: Phone, password: String): Completable  
  
fun timerFor(phone: Phone): Observable<Long>  
  
    fun getKnownPhone(): Single<Phone>  
  
fun onCountrySelect(country: Country): Completable  
  
}
```

```
interface ChangePhoneCountryUseCase {
```

```
    fun execute(): Single<Country>
```

```
}
```

```
fun getCountryByKnownPhone(): Single<Country>
```



```
fun requestOtp(phone: Phone): Completable
```

```
interface ChangePhoneCountryUseCase {  
    fun execute(): Single<Country>  
}
```

```
interface RequestOtpUseCase {  
    fun execute(): Completable  
}
```





```
fun createSession(phone: Phone, password: String): Completable
```

```
interface ChangePhoneCountryUseCase {  
  
    fun execute(): Single<Country>  
}
```

```
interface RequestOtpUseCase {  
  
    fun execute(): Completable  
}
```

```
interface CreateSessionUseCase {  
  
    fun execute(phone: Phone, password: String): C  
}
```



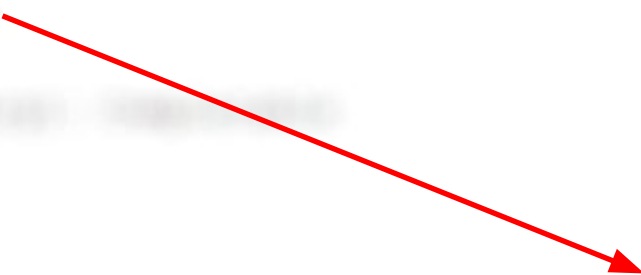
```
fun getKnownPhone(): Single<Phone>
```

```
interface ChangePhoneCountryUseCase {  
    fun execute(): Single<Country>  
}
```

```
interface RequestOtpUseCase {  
    fun execute(): Completable  
}
```

```
interface CreateSessionUseCase {  
    fun execute(phone: Phone, password: String): C  
}
```

```
interface GetUserPhoneUseCase {  
    fun execute(): Completable  
}
```

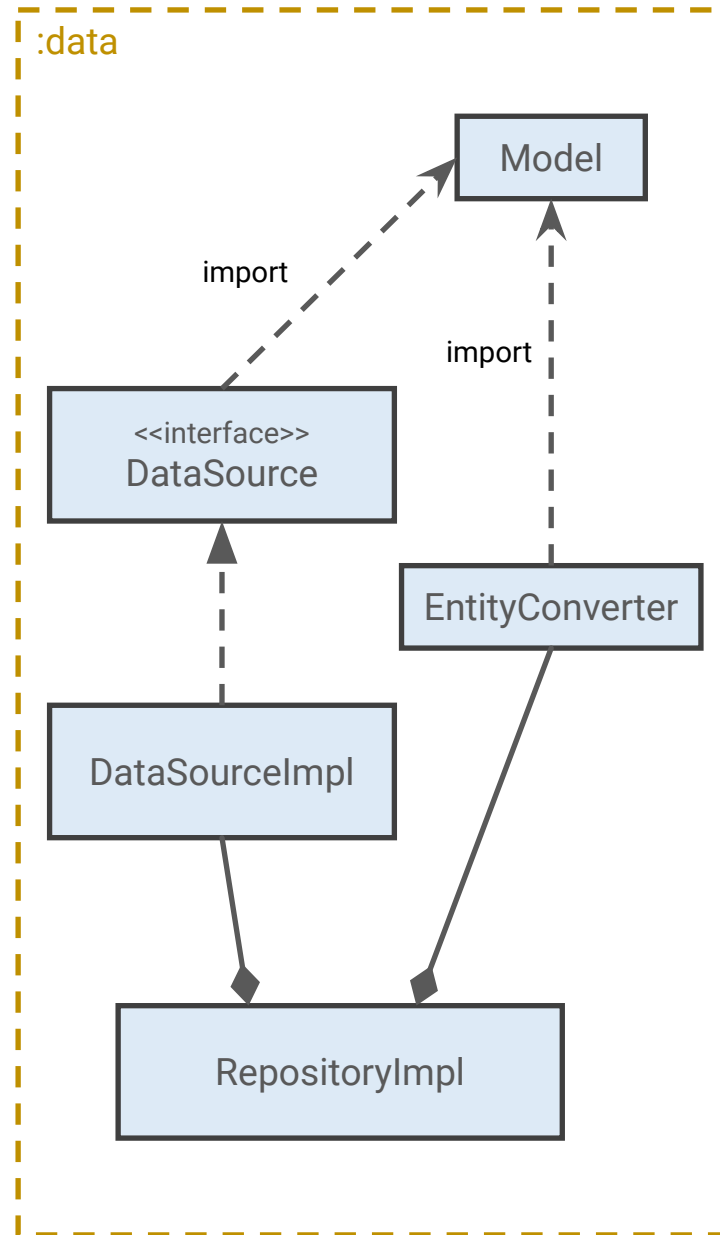


```
class LoginPresenter @Inject constructor(  
  private val loginInteractor: LoginInteractor,  
  private val router: Router) {
```

```
class LoginPresenter @Inject constructor(  
  private val changePhoneCountryUseCase: ChangePhoneCountryUseCase,  
  private val requestOtpUseCase: RequestOtpUseCase,  
  private val createSessionUseCase: CreateSessionUseCase,  
  private val getUserPhoneUseCase: GetUserPhoneUseCase,  
  private val router: Router) {
```

**Грабли №2**

**Бойлерплейт с конвертерами**



```
data class SessionModel(val token: Token,  
                        val profile: UserProfileModel,  
                        val primary: Boolean?)
```

```
class SessionConverter @Inject constructor(  
    private val userProfileConverter: TwoSideConverter<UserProfileModel, UserProfile>  
) : Converter<SessionModel, Session> {  
  
    override fun convert(from: SessionModel) = Session(  
        from.token,  
        userProfileConverter.convert(from.profile),  
        primary: from.primary ?: false  
    )  
  
}
```

```
data class Session(val token: Token,  
                  val userProfile: UserProfile,  
                  val primary: Boolean)
```

```
data class SessionModel(val token: Token,  
                        val profile: UserProfileModel,  
                        val primary: Boolean?)
```

```
override fun convert(from: SessionModel) = Session(  
    from.token,  
    userProfileConverter.convert(from.profile),  
    primary: from.primary ?: false  
)
```

```
data class Session(val token: Token,  
                  val userProfile: UserProfile,  
                  val primary: Boolean)
```



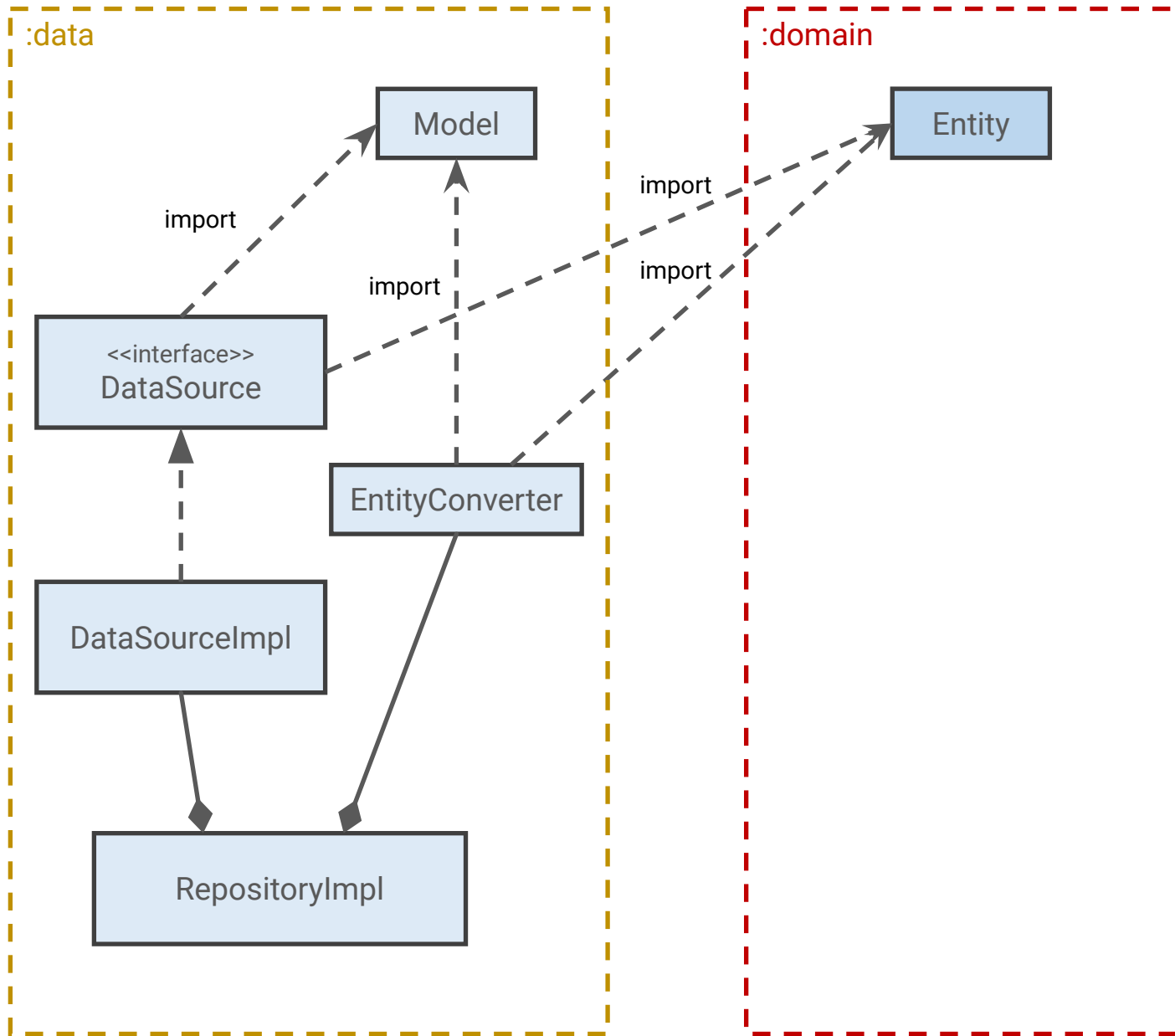
- AgentConverter
- AlphabetConverter
- AndroidLocationConverter.kt
- CardConverter
- CardTransactionConverter
- ConfigConverter
- ConfigModelConverter
- Converter.kt
- CountryToStringConverter
- CreditingCardConverter
- CurrencyConversionRateConverter
- DayDateConverter
- EnumConverter
- EnumToLowerCamelCaseStringConverter
- EventFeatureConfigConverter
- EventModelConverter
- FeatureConfigConverter
- IdentityCardConverter
- ListConverter
- MaskedUrlConfigToStringConverter
- PaymentConverter
- PaymentSystemConverter
- PersonConverter
- PersonToUserProfileModelConverter
- PromoActionConverter
- PromoCodeConverter
- RatingConfigConverter
- ResultConverter
- SaveCardDataConverter
- SessionConverter
- TariffAlternativeConverter

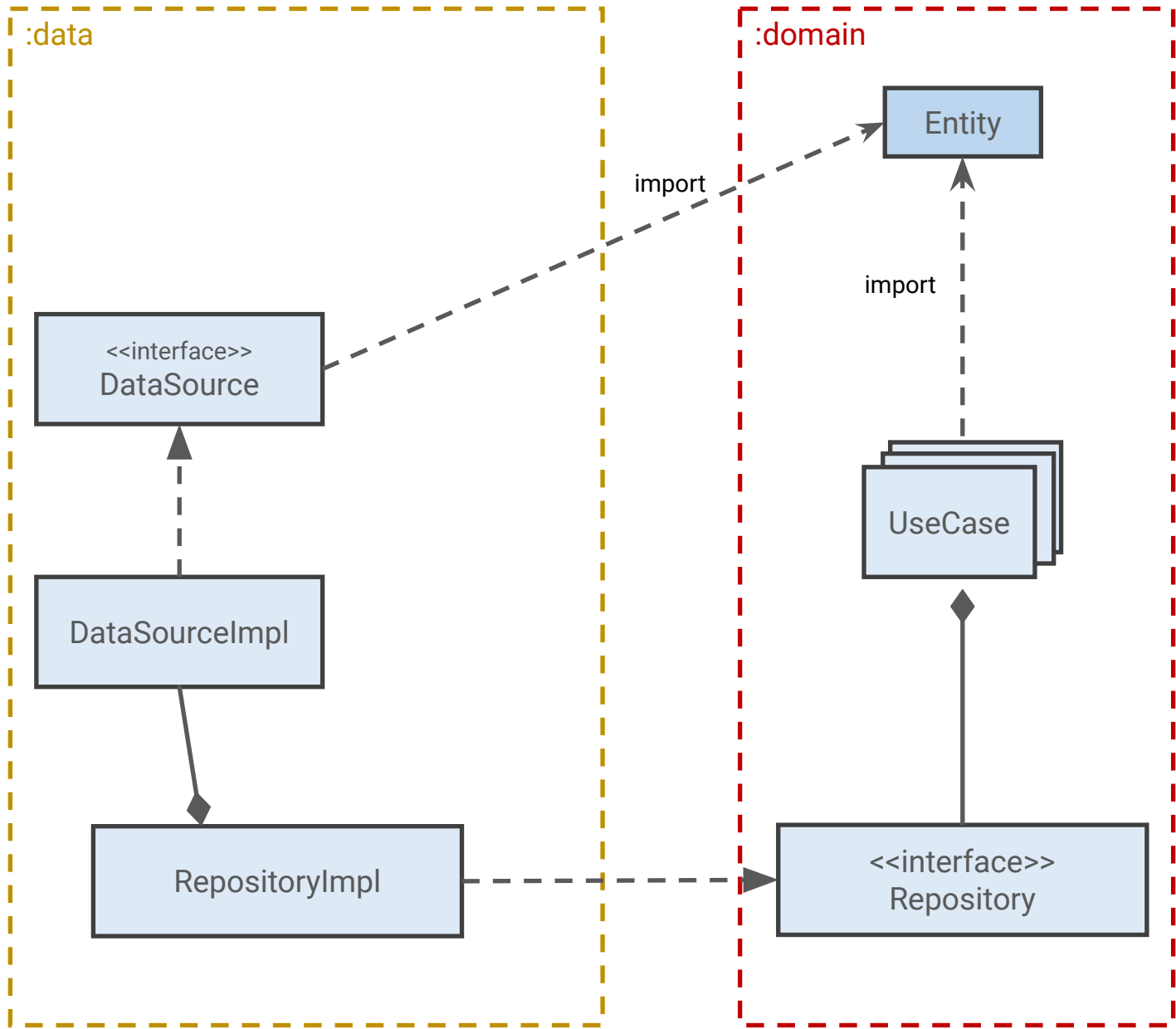




**Решение**

**Убираем модели и конвертеры**





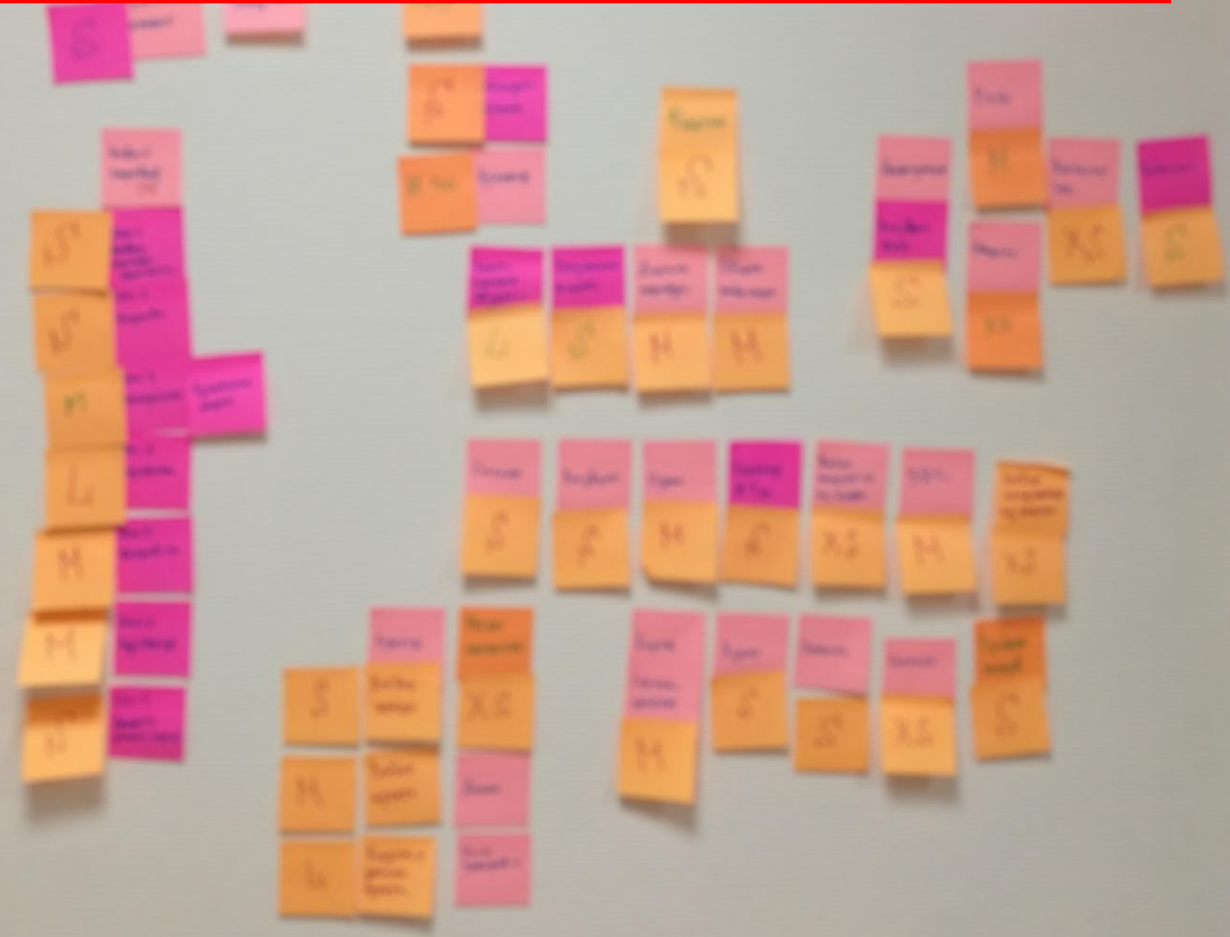
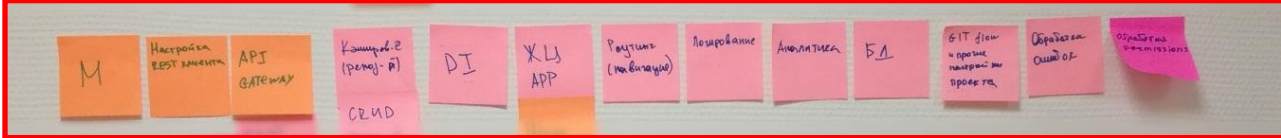
# Выбираем технологии



# Не тратить время на изобретение велосипедов



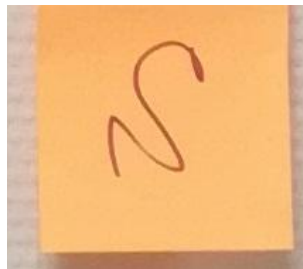






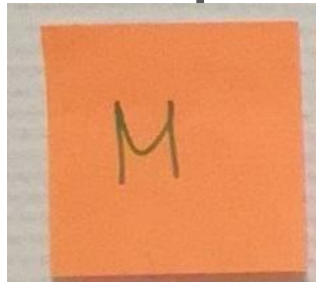
?

Dagger2



или

Toothpick



Настройка  
REST клиента

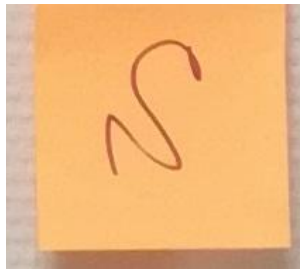
API  
GATEWAY

DI

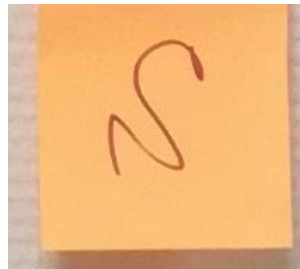
БД



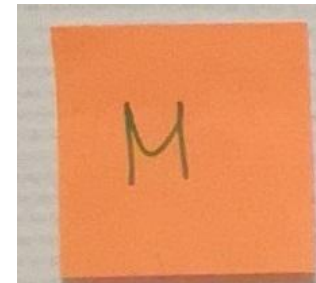
Retrofit



Dagger2



Room





***“Если хочешь рассмешить Бога,  
расскажи ему о своих планах”***



# Проблема №1

**Dagger2**

```
interface AppComponent {  
  
    fun getSplashPresenter(): SplashPresenter  
  
    fun getLanguagePresenter(): LanguagePresenter  
  
    fun getForceUpdatePresenter(): ForceUpdatePresenter  
  
    fun getPushInteractor(): PushInteractor  
  
    fun getPushNotificationPresenter(): PushNotificationPresenter  
  
    fun getEda(): Eda  
  
    fun getEdaInteractor(): EdaInteractor  
  
    fun plusLoginComponent(module: LoginModule): LoginComponent  
  
    fun plusSessionComponent(): SessionComponent  
  
    fun app(): App  
  
    fun context(): Context  
  
    fun inject(splashActivity: SplashActivity)  
  
    fun inject(launchActivity: LaunchActivity)  
  
    fun inject(loginFragment: LoginFragment)  
  
    fun inject(languageActivity: LanguageActivity)
```



**Решение**

**AndroidInjector, @Binds и @Contributes**

```
@Module(includes = [
    AndroidSupportInjectionModule::class,
    RouterModule::class,
    ValidationModule::class,
    SessionModule::class])
abstract class AppModule {

    @Binds
    @AppScope
    abstract fun bindApplicationIdCreator(creator: ApplicationIdCreatorImpl): ApplicationIdCreator

    @ActivityScope
    @ContributesAndroidInjector(modules = [
        LaunchActivityModule::class,
        LoginFragmentModule::class,
        HistoryFragmentModule::class,
        AgentsFragmentModule::class,
        HelpFragmentModule::class,
        SettingsFragmentModule::class,
        NonEditableDetailsLaunchFragmentModule::class,
        ErrorHandlerModule::class,
        TransferDetailFragmentModule::class,
        RefundFragmentsModule::class,
        PromoFragmentModule::class,
        LimitsFragmentModule::class,
        EventInfoFragmentModule::class,
        ReceiversSettingsFragmentModule::class,
        EmailTransferFragmentModule::class
    ])
    abstract fun provideLaunchActivity(): LaunchActivity
}
```

@Binds

@AppScope

**abstract fun** bindApplicationIdCreator(creator: ApplicationIdCreatorImpl): ApplicationIdCreator

@AppScope

@ActivityScope



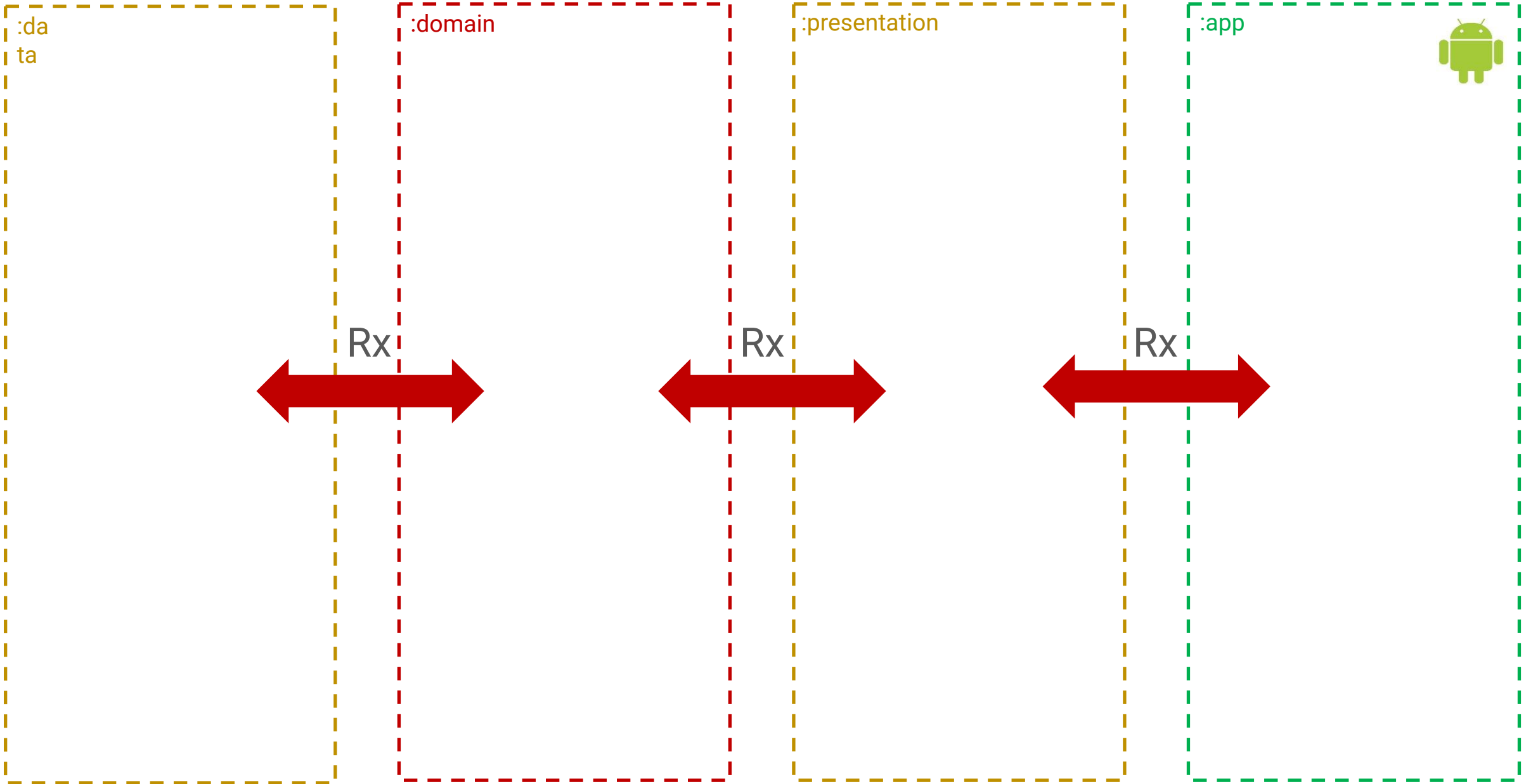
```
@ActivityScope
@ContributesAndroidInjector(modules = [
    LaunchActivityModule::class,
    LoginFragmentModule::class,
    HistoryFragmentModule::class,
    AgentsFragmentModule::class,
    HelpFragmentModule::class,
    SettingsFragmentModule::class,
    NonEditableDetailsLaunchFragmentModule::class,
    ErrorHandlerModule::class,
    TransferDetailFragmentModule::class,
    RefundFragmentsModule::class,
    PromoFragmentModule::class,
    LimitsFragmentModule::class,
    EventInfoFragmentModule::class,
    ReceiversSettingsFragmentModule::class,
    EmailTransferFragmentModule::class
])
abstract fun provideLaunchActivity(): LaunchActivity
```

```
}
```



# Проблема №2

**Rx Hell**



```
interface ChangePhoneCountryUseCase {  
  
    fun execute(): Single<Country>  
}
```

```
class RxPreferences @Inject constructor(context: Context) {  
  
    companion object {  
        private const val SHARED_PREFERENCES_NAME = "SHARED_PREFERENCES_NAME"  
    }  
  
    private val preferences: SharedPreferences = context.getSharedPreferences(SHARED_PREFERENCES_NAME, Context.MODE_PRIVATE)  
  
    fun getStringWithDefault(key: String, default: String = emptyString()): Single<String> =  
        Single.fromCallable { preferences.getString(key, default) }  
  
    fun getBooleanWithDefault(key: String, default: Boolean = false): Single<Boolean> =  
        Single.fromCallable { preferences.getBoolean(key, default) }  
  
    fun getIntWithDefault(key: String, default: Int = 0): Single<Int> =  
        Single.fromCallable { preferences.getInt(key, default) }  
  
    fun getLongWithDefault(key: String, default: Long = 0L): Single<Long> =  
        Single.fromCallable { preferences.getLong(key, default) }
```

```
Single.fromCallable { preferences.getString(key, default) }
```

```
Single.fromCallable { preferences.getBoolean(key, default) }
```

```
Single.fromCallable { preferences.getInt(key, default) }
```

```
Single.fromCallable { preferences.getLong(key, default) }
```

**Еще один пример**

```
RxView.focusChanges(editText)
    .map { hasFocus ->
        if (hasFocus) {
            editText.setSelection(editText.text.length)
        }
        ^map hasFocus
    }
    .filter { !it }
    .map { it: Boolean
        if (trimOnFocusLost) {
            editText.setText(editText.text.trim())
        }
        ^map editText.editableText.toString()
    }
    .skip(count: 1)
    .subscribeOver { it: String!
        if (!focusChangeMuted) {
            onFocusLost(it)
        }
        focusChangeMuted = false
    }
    .addTo(disposable)
```

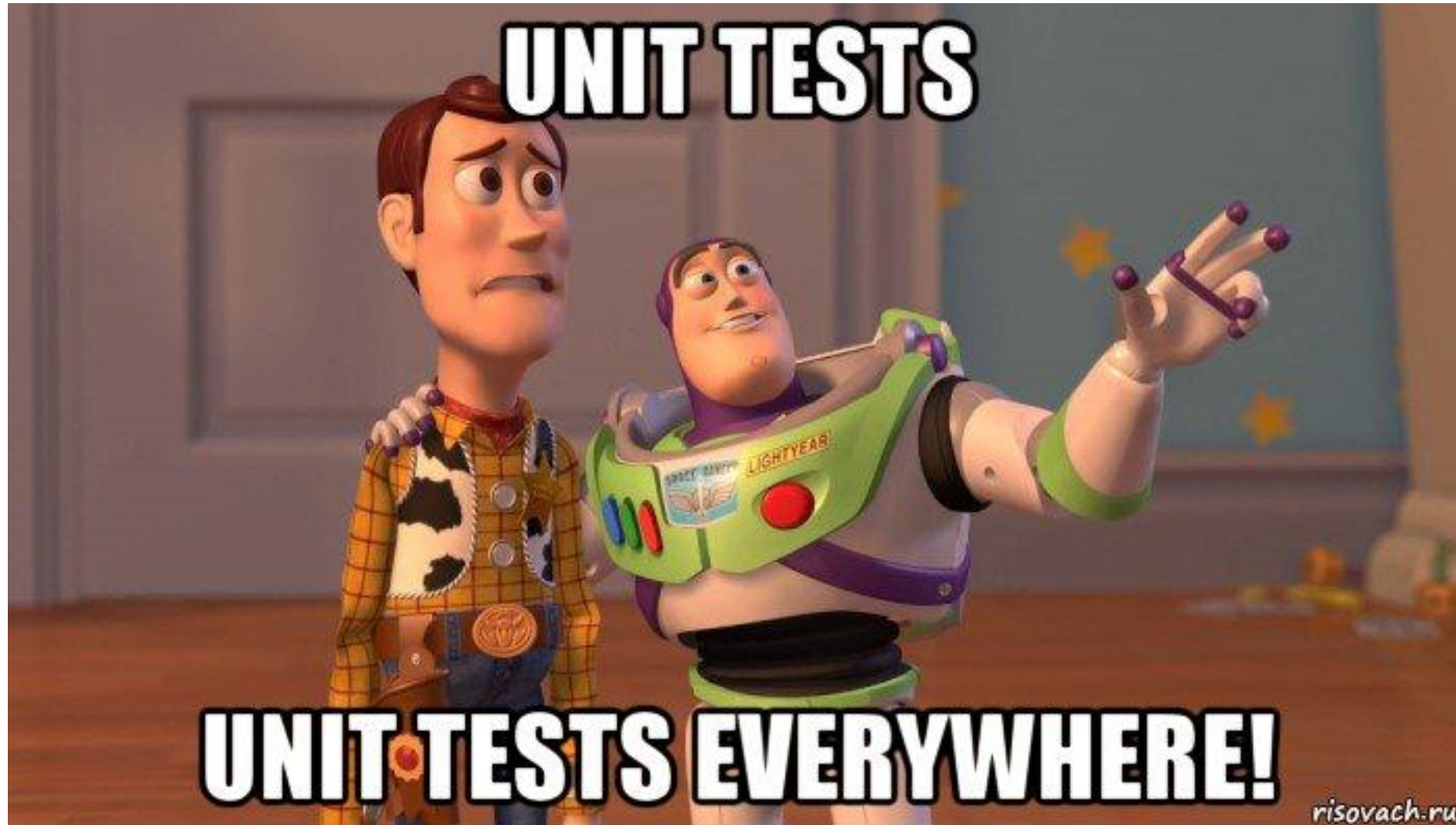


# Правило

**«Логика использования технологии  
не должна быть сложнее  
логики решения задачи»**



# Тесты нам помогают



**Не принимаем код без Unit-тестов**



Projects

Changes

Agents



Build Queue

Item / Commit/Revision

#7677 (07 Sep 18 15:17)

Overview

Changes

Tests

Build Log

Parameters

Artifacts



Improvement/QAS-785 (3.7.0)

Создан Анна Саврасова - #620, last



QAS-841 [PvF] рефакторинг рк

Создан Анна Саврасова - #623,



test/QAS-537 - зграв рграв

Создан Анна Саврасова

Result:

❗ Tests failed: 1, passed: 1703

Time:

07 Sep 18 15:17 - 15:21 (4m:01s)

Branch:

838

Q 9



+2



Q 4



+2



Q 82



+3



❗ Build problems: 1

❗ Process exited with code 1 (Step: Grails) ❗

❗ 1 test failed (see above)

❗ 1703 tests passed (all tests)

1703 tests passed (all tests)

1703 tests passed (all tests)



Projects [+]

Changes

Agents [3]



Build Queue [3]

[[ Rang / [ Commit/Verification

Improvement/QAS-71  
Carpenter Andre Caprianni

QAS-841 [PvF] perbar  
Carpenter Andre Caprianni

test/QAS-537 - wq  
Bismillah Prince Waj

#7681 (07 Sep 18 15:33)

Overview

Changes [3]

Tests

Build Log

Parameters

Artifacts

[Avatar] [Avatar] +2 [Checkmark]

[Avatar] [Avatar] +2 [Warning]

[Avatar] [Avatar] +3 [Checkmark]

Result:

✓ Tests passed: 1741

Time:

07 Sep 18 15:33 - 15:42 (9m:12s)

Branch:

841

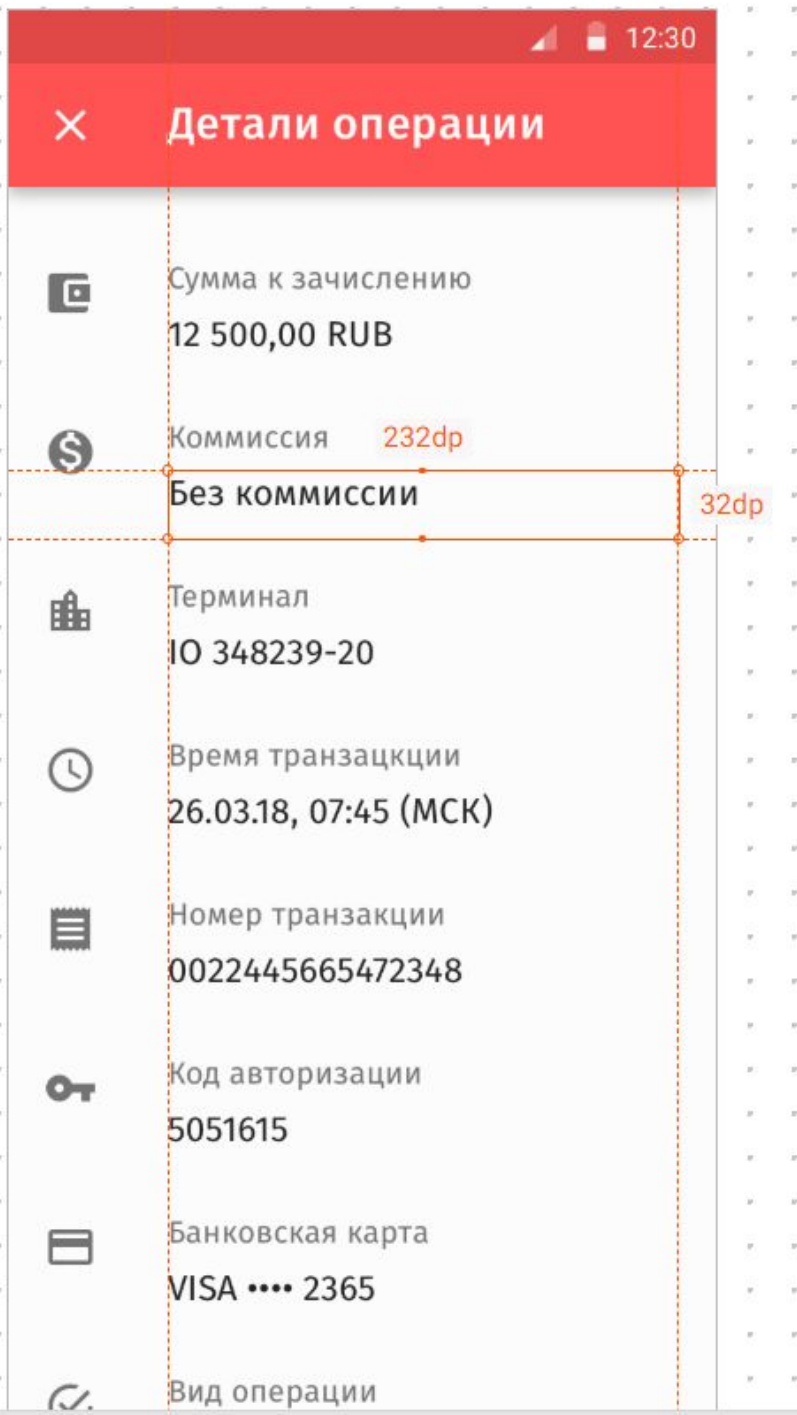
1741 tests passed (all tests)





# В одной упряжке с дизайнерами








# Color Palette

 **#Red\_A200**  
#ff5252 100%  
rgb 255 82 82

 **#Red\_A100**  
#ff8a80 100%  
rgb 255 138 128

 **#Blue\_A100**  
#82b1ff 100%  
rgb 130 177 255

 **#Blue\_A700**  
#2962ff 100%  
rgb 41 98 255

 **#Red\_900**  
#b71c1c 100%  
rgb 183 28 28

 **#Yellow\_700**  
#fbc02d 100%  
rgb 251 192 45

 **#Black\_87**  
#000000 87%  
rgb 0 0 0

 **#Black\_54**  
#000000 54%  
rgb 0 0 0

 **#Black\_38**  
#000000 38%  
rgb 0 0 0

 **#white\_100**  
#ffffff 100%  
rgb 255 255 255

 **#white\_70**  
#ffffff 70%  
rgb 255 255 255

 **#white\_30**  
#ffffff 30%  
rgb 255 255 255

```
<!-- Color palette -->  
  
<resources>  
  <color name="red__a200">#ff5252</color>  
  <color name="red__a100">#ff8a80</color>  
  <color name="blue__a100">#82b1ff</color>  
  <color name="red__a700">#d50000</color>  
  <color name="blue__a200">#448aff</color>  
  <color name="blue__a700">#2962ff</color>  
  <color name="red_900">#b71c1c</color>  
  <color name="yellow_700">#fbc02d</color>  
  <color name="green_500">#4caf50</color>  
  <color name="blue__a400">#2979ff</color>  
  <color name="black_87">#de000000</color>  
  <color name="black_54">#8a000000</color>  
  <color name="black_38">#61000000</color>  
  <color name="black_24">#3d000000</color>  
  <color name="black_12">#1e000000</color>  
  <color name="white_100">#ffffff</color>  
  <color name="white_70">#b3ffffff</color>  
  <color name="white_30">#4cffffff</color>  
  <color name="white_12">#1effffff</color>  
</resources>
```



# Display

Headline

FiraSans, Regular, 36sp  
Line height: 48sp, Letter spacing: 0.1

FiraSans, Regular, 24sp  
Line height: 32sp, Letter spacing: -0.1

Title

FiraSans, Medium, 20sp  
Line height: 28sp, Letter spacing: 0.4

Subheader

FiraSans, Regular, 18sp  
Line height: 24sp, Letter spacing: 0.1

Body 2

FiraSans, Medium, 16sp  
Line height: 20sp, Letter spacing: 0.2

Button

FiraSans, Medium, 16sp  
Line height: 20sp, Letter spacing: 0.4

Body 1

FiraSans, Regular, 16sp  
Line height: 20sp, Letter spacing: 0.1

Caption

FiraSans, Regular, 12sp  
Line height: 16sp, Letter spacing: 0.1

```
<style name="Body2">  
  <item name="android:textSize">16sp</item>  
  <item name="android:letterSpacing">0.2</item>  
  <item name="android:lineSpacingExtra">8sp</item>  
</style>
```

```
<style name="Headline">  
  <item name="android:textSize">24sp</item>  
  <item name="android:letterSpacing">0</item>  
  <item name="android:lineSpacingExtra">8sp</item>  
</style>
```

```
<style name="Title">  
  <item name="android:textSize">20sp</item>  
  <item name="android:letterSpacing">0.4</item>  
  <item name="android:lineSpacingExtra">8sp</item>  
</style>
```

```
<style name="Subheader">  
  <item name="android:textSize">18sp</item>  
  <item name="android:letterSpacing">0.1</item>  
  <item name="android:lineSpacingExtra">8sp</item>  
</style>
```

# Не попадаем в дедлайн



# **Строим итеративный план спринтов**

<b>Команда</b>
Ваня
Джон
Кирилл
Гриша
Вася

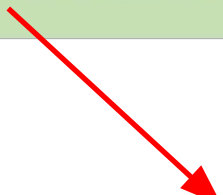
Команда	MT-Android 2.14				
	неделя 1				
	04.12	05.12	06.12	07.12	08.12
Ваня					
Джон					
Кирилл					
Гриша					
Вася					

Даты

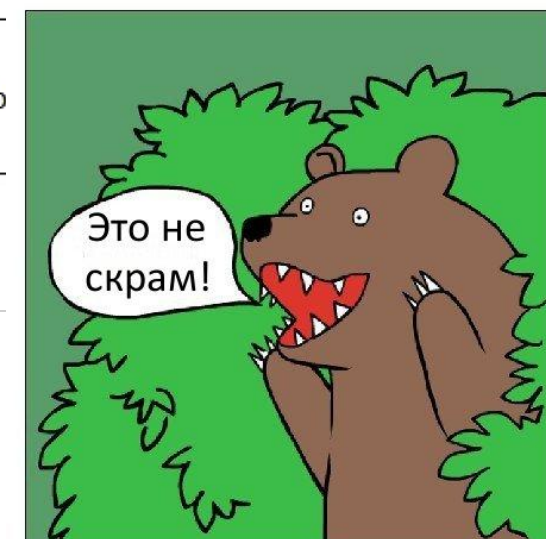


Команда	MT-Android 2.14				
	неделя 1				
	04.12	05.12	06.12	07.12	08.12
Ваня	Новый перевод. Калькулятор - доделки			Навигация вперед	Выбор эмитента - интеграция с др. компонентами
Джон					
Кирилл					
Гриша					
Вася					

Задачи



Команда	MT-Android 2.14				
	неделя 1				
	04.12	05.12	06.12	07.12	08.12
Ваня	Новый перевод. Калькулятор - доделки			Навигация вперед	Выбор эмитента - интеграция с др. компонентами
Джон	Новый перевод. Получатель - domain	Новый перевод. Получатель - pres			
Кирилл				Возвраты и переход на API 2.0 в старом Д	
Гриша	Новый перевод. Оплата - data	Новый перевод. Оплата - domain	Новый перевод. Оплата - presentation		Но
Вася	Новый перевод. Получатель - data	Новый перевод. Отправитель - domain	Новый перевод. Отправитель - data	Новый перевод. Отправитель - presentation	





**Но команде прозрачно  
и бизнесу спокойно**





# Эпилог

*“Если головоломка не сложилась,  
и тебе уже не собрать пазлы  
— начни сначала” © Death Note*

**Переписать приложение с нуля  
и не потерпеть фиаско – можно!**

**Но для этого...**

# **Задать вопрос – «А нужно ли?»**

**Собрать команду единомышленников и оценить масштаб**

**Построить архитектуру решающую задачи  
бизнеса**

**Выбрать технологии, основываясь на опыте  
команды**

**Использовать Unit-тесты, они сэкономят время на  
отладке**

**Дизайнеры могут помочь, для этого есть все  
инструменты**

Задать вопрос – «А нужно ли?»

**Собрать команду единомышленников и оценить масштаб**

Построить архитектуру решающую задачи бизнеса

Выбрать технологии, основываясь на опыте команды

Использовать Unit-тесты, они сэкономят время на отладке

Дизайнеры могут помочь, для этого есть все инструменты

Задать вопрос – «А нужно ли?»

Собрать команду единомышленников и оценить масштаб

**Построить архитектуру решающую задачи бизнеса**

Выбрать технологии, основываясь на опыте команды

Использовать Unit-тесты, они сэкономят время на отладке

Дизайнеры могут помочь, для этого есть все инструменты

Задать вопрос – «А нужно ли?»

Собрать команду единомышленников и оценить масштаб

Построить архитектуру решающую задачи бизнеса

**Выбрать технологии, основываясь на опыте команды**

Использовать Unit-тесты, они сэкономят время на отладке

Дизайнеры могут помочь, для этого есть все инструменты

Задать вопрос – «А нужно ли?»

Собрать команду единомышленников и оценить масштаб

Построить архитектуру решающую задачи бизнеса

Выбрать технологии, основываясь на опыте команды

**Использовать Unit-тесты, они сэкономят время на отладке**

Дизайнеры могут помочь, для этого есть все инструменты

Задать вопрос – «А нужно ли?»

Собрать команду единомышленников и оценить масштаб

Построить архитектуру решающую задачи бизнеса

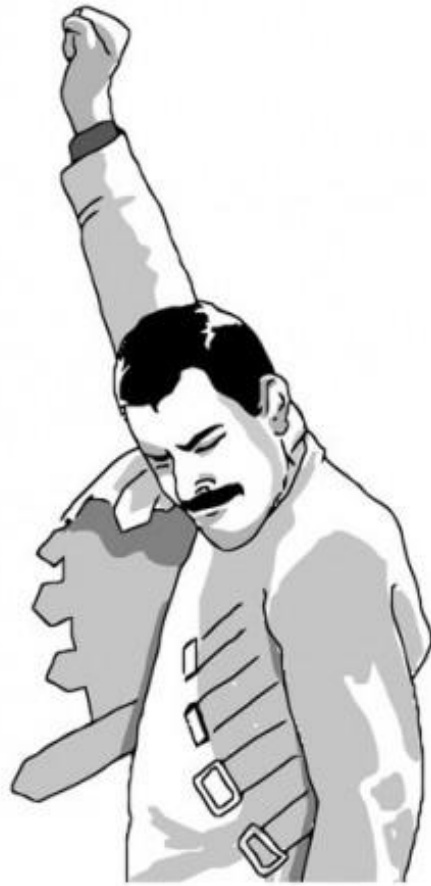
Выбрать технологии, основываясь на опыте команды

Использовать Unit-тесты, они сэкономят время на отладке

**Дизайнеры могут помочь, для этого есть все инструменты**



**Нам это удалось,  
чего и вам желаем!**



**Спасибо** 😊

Емельянов Михаил,  
[m.emeljanov@cft.ru](mailto:m.emeljanov@cft.ru)