Как переписать приложение с нуля и не потерпеть фиаско





Осебе

7 лет в Android-разработке

В ЦФТ с 2013 г.

Разработал более десятка проектов











- 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

- 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

- 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

- 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

- 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

- 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?

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

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

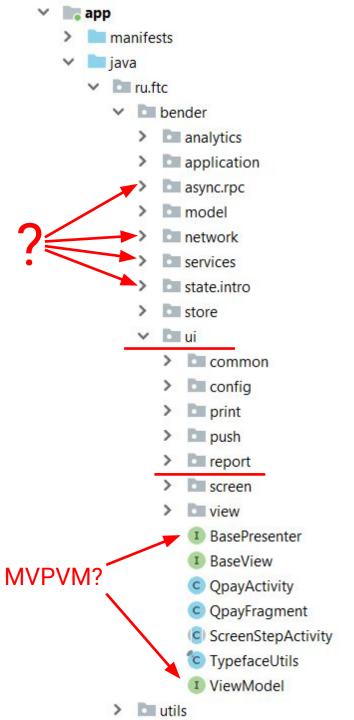
- Масштабироваться
- Увеличить скорость разработки фич
- Новый дизайн
- Улучшить стабильность

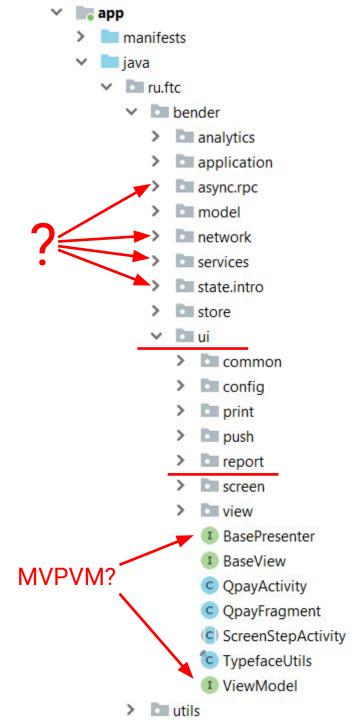
- Масштабироваться
- Увеличить скорость разработки фич
- Новый дизайн
- Улучшить стабильность

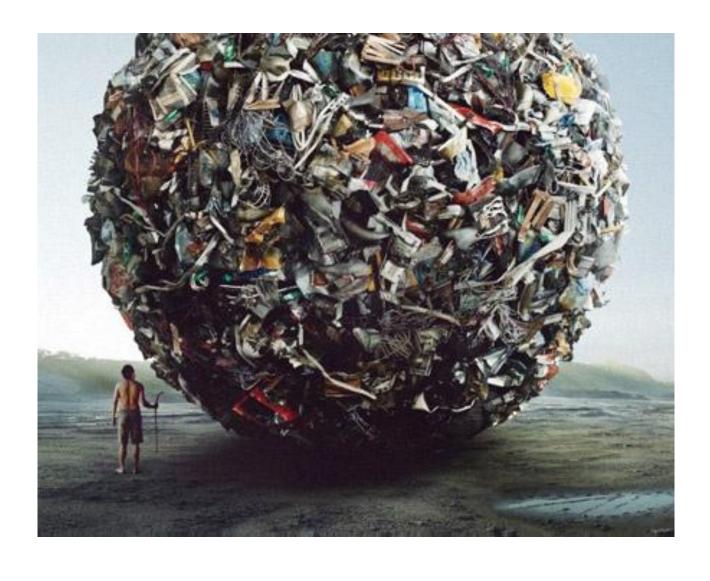
Но не все так просто...

Проблема №1

Архитектура







```
class FragmentPaymentViewModel extends FragmentPaymentViewModelPartial implements QpayWebViewClient.OnLoading
    private UrlsWrapper urls;
    private final QpayWebViewClient webViewClient;
    private OnPaymentActionSuccessListener onPaymentActionSuccesListener;
    private final BaseFragment fragment;
    FragmentPaymentViewModel(final Bundle state, final BaseFragment fragment, final UrlsWrapper urls) {
        super(state, fragment.getActivity());
        this.fragment = fragment;
```

```
getWebView().getSettings().setSupportMultipleWindows(true);
getWebView().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);
                                                        Стартуем Activity
        return false;
});
                                                    Делаем операции View
setWebClient (webViewClient);
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));
                                                    Отправляем broadcast
            return true;
          else {
            return false;
1);
```

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
```

```
<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">#FFFFF66D</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



В результате получили

Нет архитектуры – кругом спагеттикод

Нет тестов – делаем одно, ломаем другое

Долгий рефакторинг – тонем в техдолге

Нет архитектуры – кругом спагеттикод

Нет тестов – делаем одно, ломаем другое

Долгий рефакторинг – тонем в техдолге

Нет архитектуры – кругом спагеттикод

Нет тестов – делаем одно, ломаем другое

Долгий рефакторинг – тонем в техдолге

Нет архитектуры – кругом спагетти- код

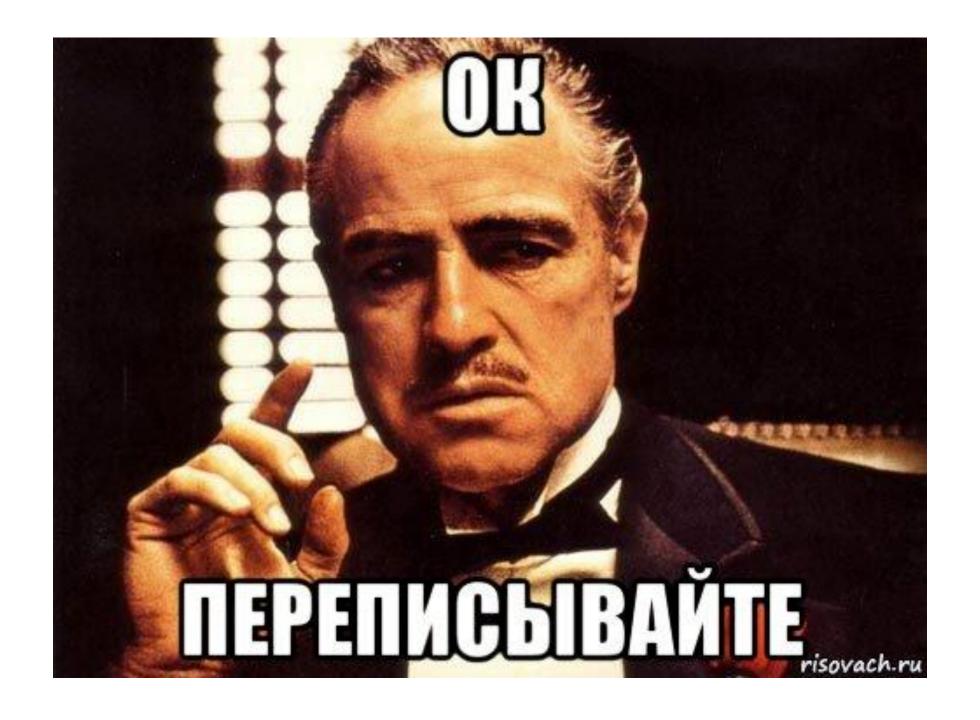
Нет тестов – делаем одно, ломаем другое

Долгий рефакторинг – тонем в техдолге

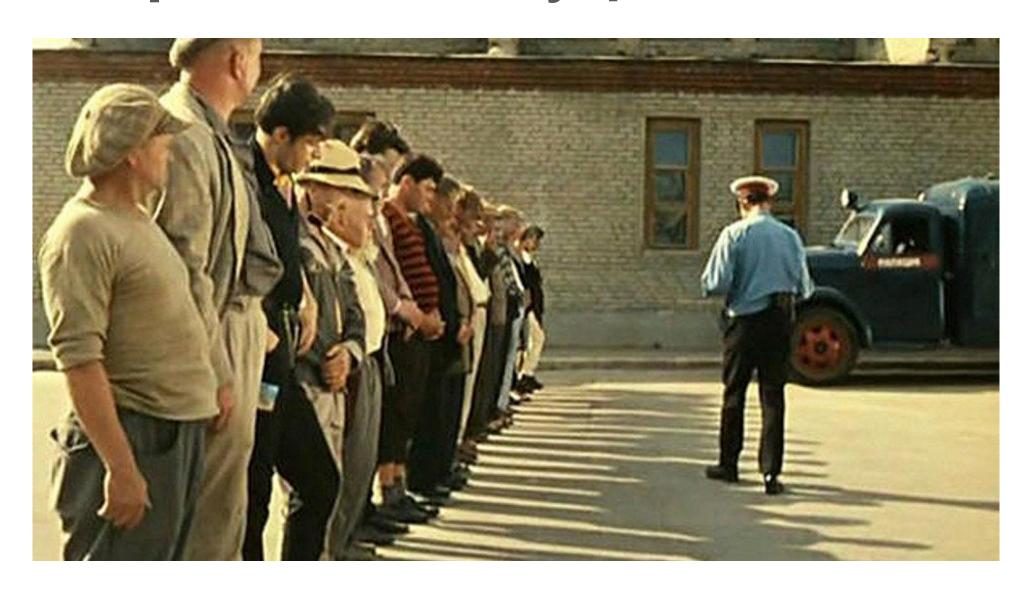
Нет архитектуры – кругом спагетти- код

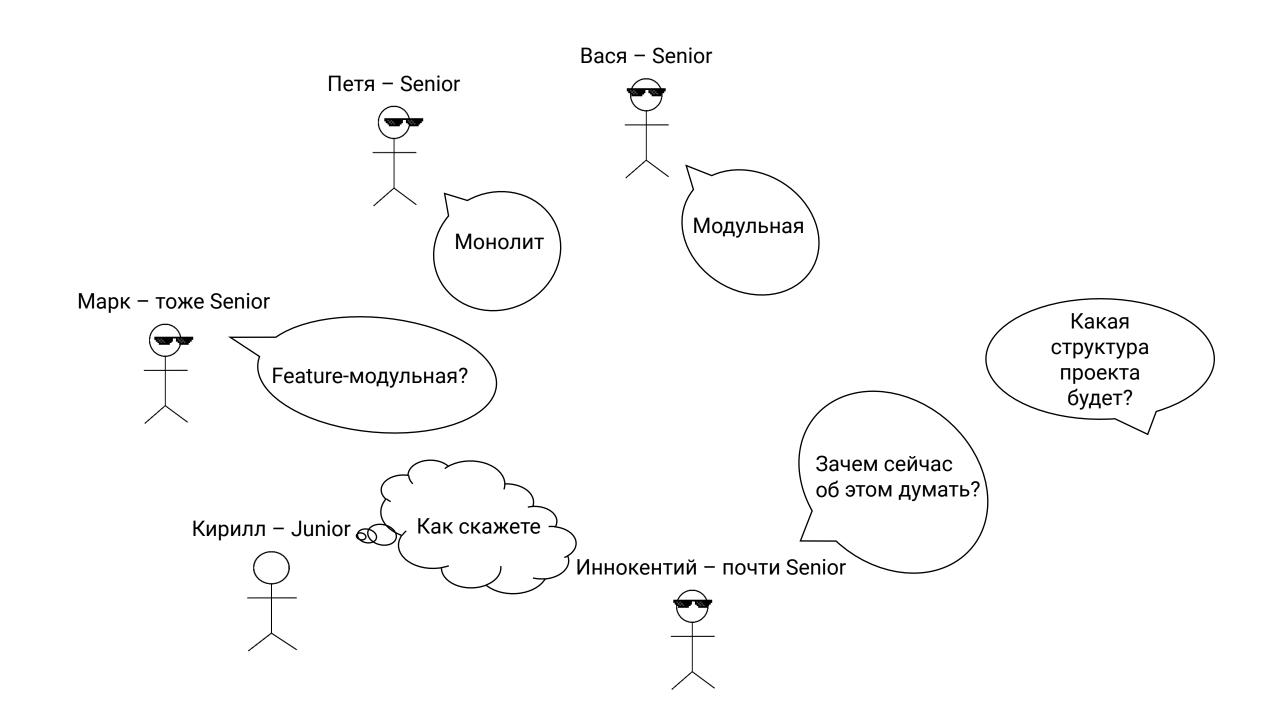
Нет тестов – делаем одно, ломаем другое

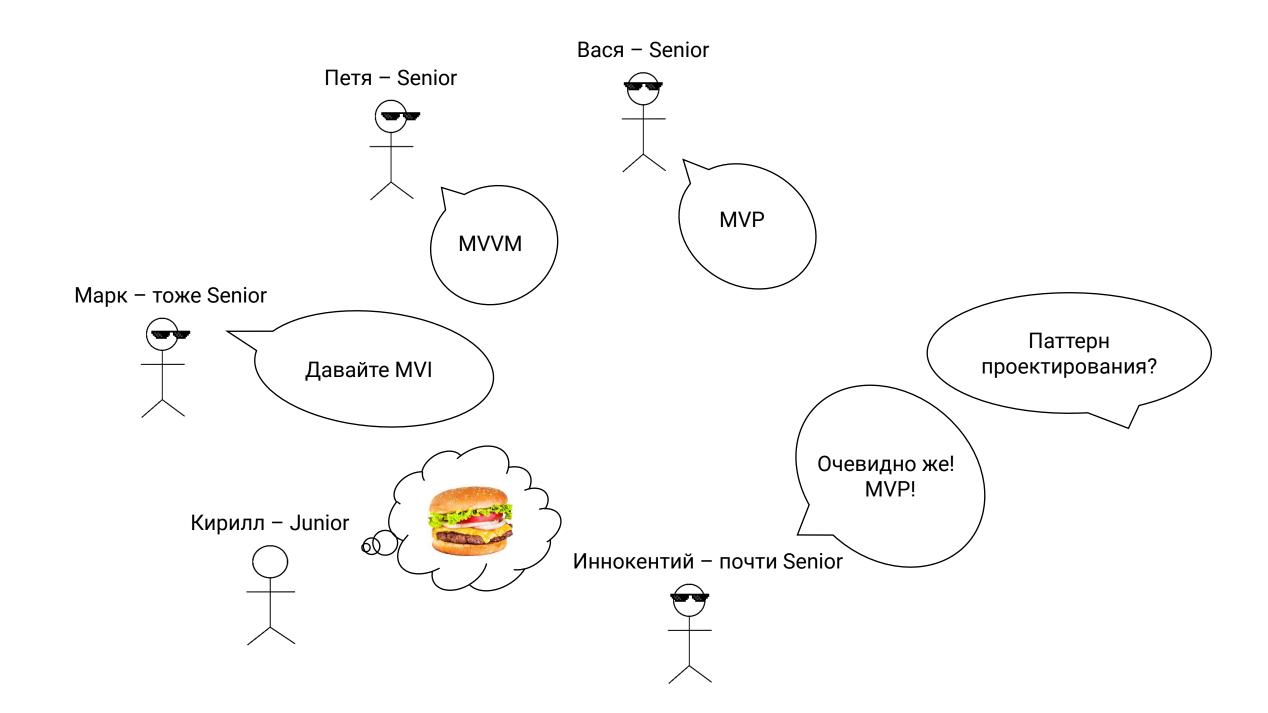
Долгий рефакторинг – тонем в техдолге



Набираем команду, решаем задачи







Прошло много часов, ни одного решения

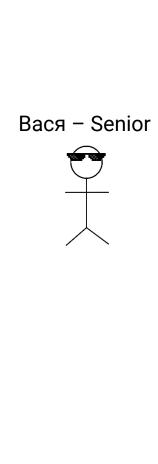


Правило «Команда из сильных разработчиков, не всегда делает разработку сильной»

Уволить всех Senior?

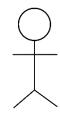
Оставить 1-го





Кирилл – Junior

Гриша – Middle



Критерии эффективной команды

1 tech lead

Уровень знаний не ниже среднего

Прагматичный выбор технологий

1 tech lead

Уровень знаний не ниже среднего

Прагматичный выбор технологий

1 tech lead

Уровень знаний не ниже среднего

Прагматичный выбор технологий

1 tech lead

Уровень знаний не ниже среднего

Прагматичный выбор технологий

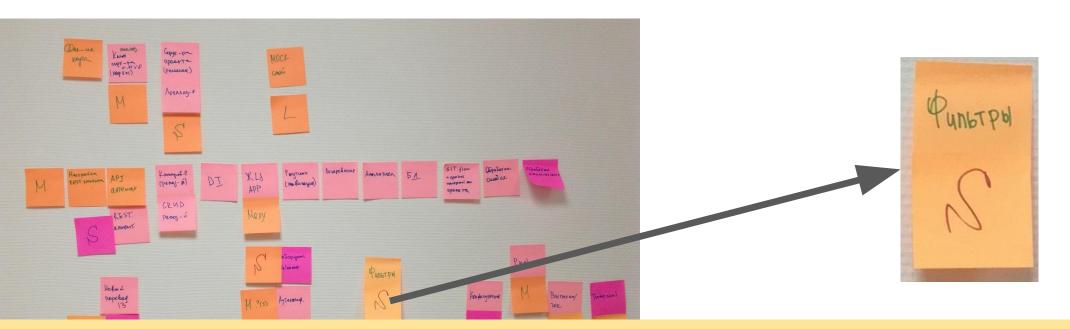
1 tech lead

Уровень знаний не ниже среднего

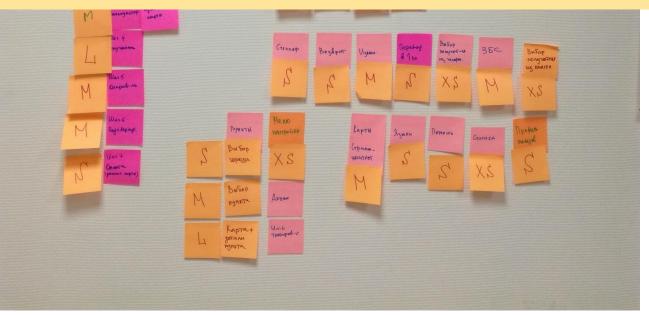
Прагматичный выбор технологий

Разбираем приложение по кирпичикам





~ 5 месяцев



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

• Считаем, отдаем бизнесу

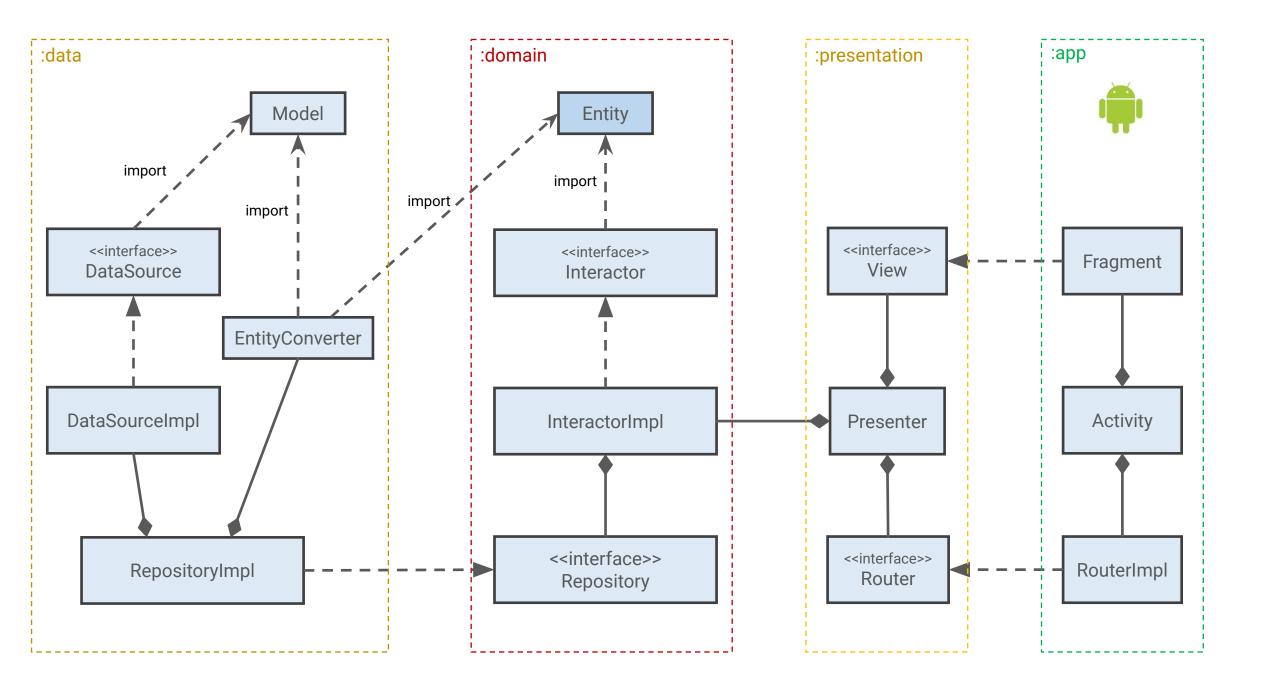
Строим архитектуру





Требования к архитектуре

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

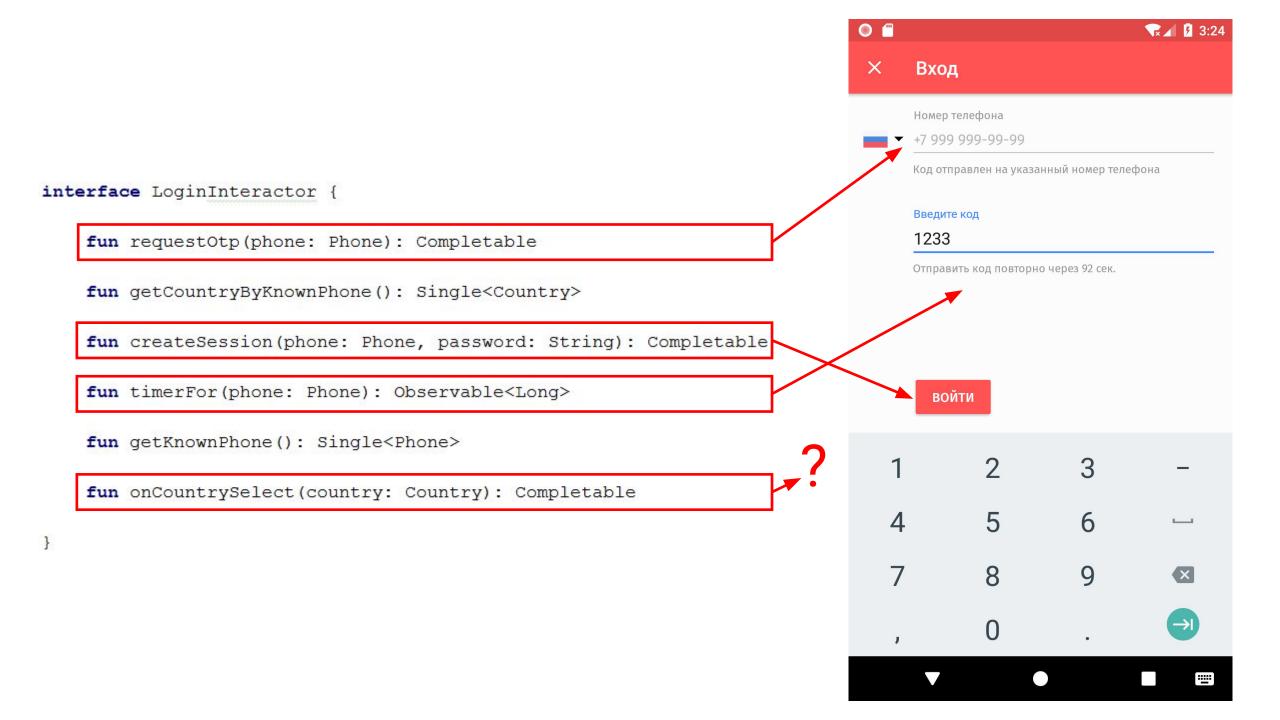


Но мы наступили на грабли:(



Грабли №1

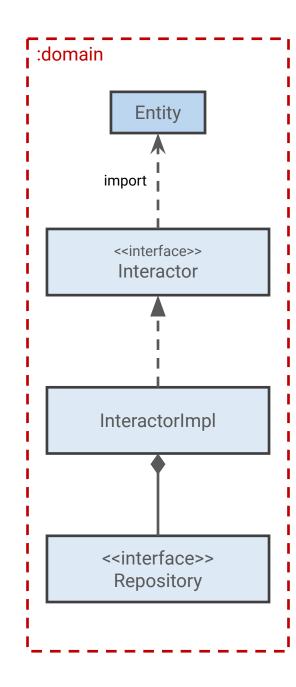
Бизнес-логика == представлению

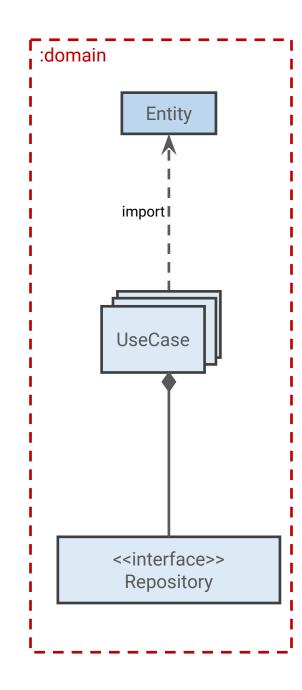


class LoginInteractorImpl @Inject constructor(

```
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 analytics: AnalyticsConfiguration
```

Переделываем на UseCase





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

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

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

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

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

```
interface ChangePhoneCountryUseCase {
                                                                   fun execute(): Single<Country>
                                                               interface RequestOtpUseCase {
                                                                   fun execute(): Completable
fun createSession(phone: Phone, password: String): Completable
                                                               interface CreateSessionUseCase {
                                                                   fun execute (phone: Phone, password: String): 0
```

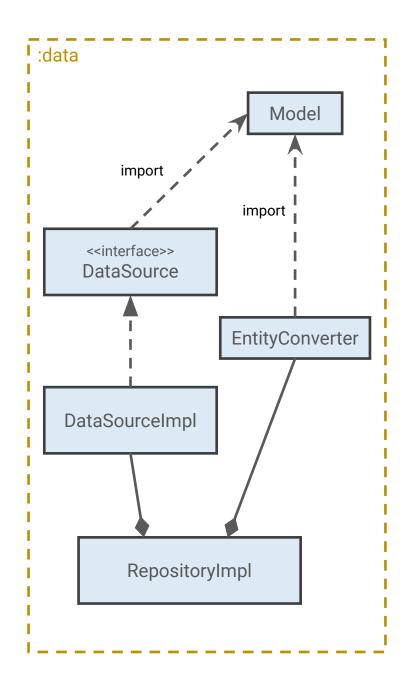
```
fun getKnownPhone(): Single<Phone>
```

```
interface ChangePhoneCountryUseCase {
    fun execute(): Single<Country>
interface RequestOtpUseCase {
    fun execute(): Completable
interface CreateSessionUseCase {
    fun execute (phone: Phone, password: String): 0
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 < Session Model, 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,

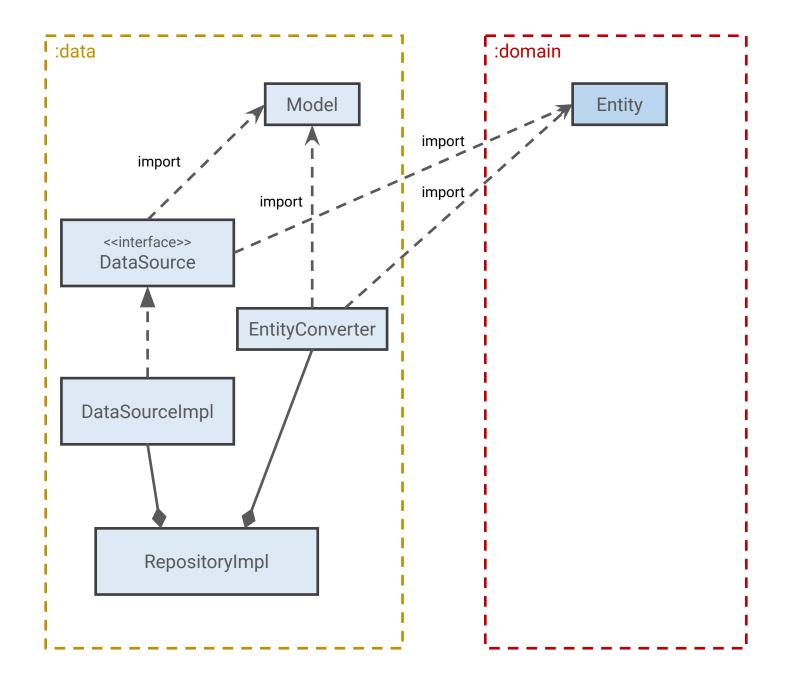
val primary: Boolean?)
```

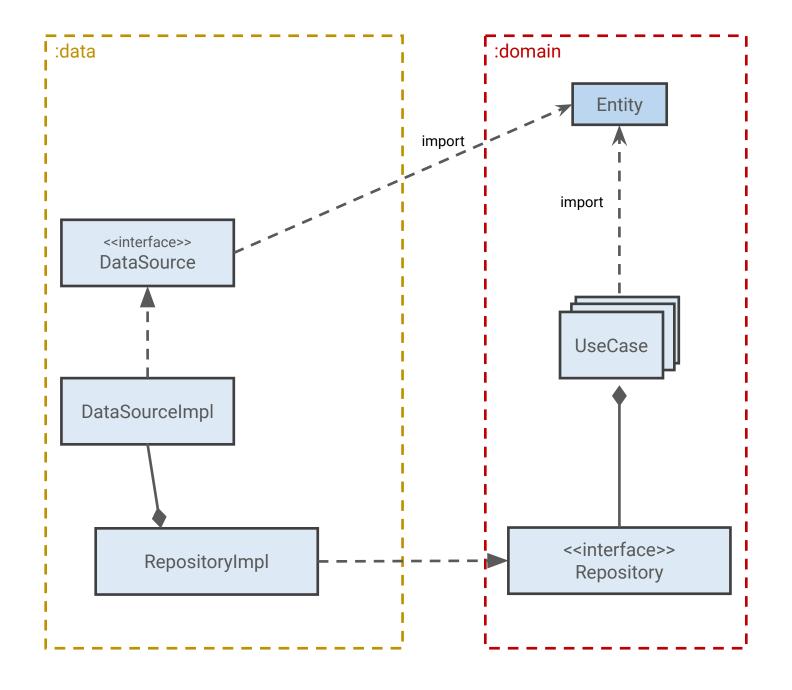
```
override fun convert(from: SessionModel) = Session(
    from.token,
    userProfileConverter.convert(from.profile),
    primary: from.primary ?: false
)
```

- ♠ AgentConverter
- AndroidLocationConverter.kt
- CardConverter
- CardTransactionConverter
- ConfigConverter
- ConfigModelConverter
- Converter.kt
- CountryToStringConverter
- CreditingCardConverter
- CurrencyConversionRateConverter
- □ DayDateConverter
- **©** EnumConverter
- EnumToLowerCamelCaseStringConverter
- EventFeatureConfigConverter
- @ EventModelConverter
- FeatureConfigConverter
- IdentityCardConverter
- C ListConverter
- MaskedUrlConfigToStringConverter
- PaymentConverter
- PaymentSystemConverter
- PersonConverter
- PersonToUserProfileModelConverter
- PromoActionConverter
- PromoCodeConverter
- RatingConfigConverter
- ResultConverter
- SaveCardDataConverter
- Session Converter
- TariffAlternativeConverter



Убираем модели и конвертеры



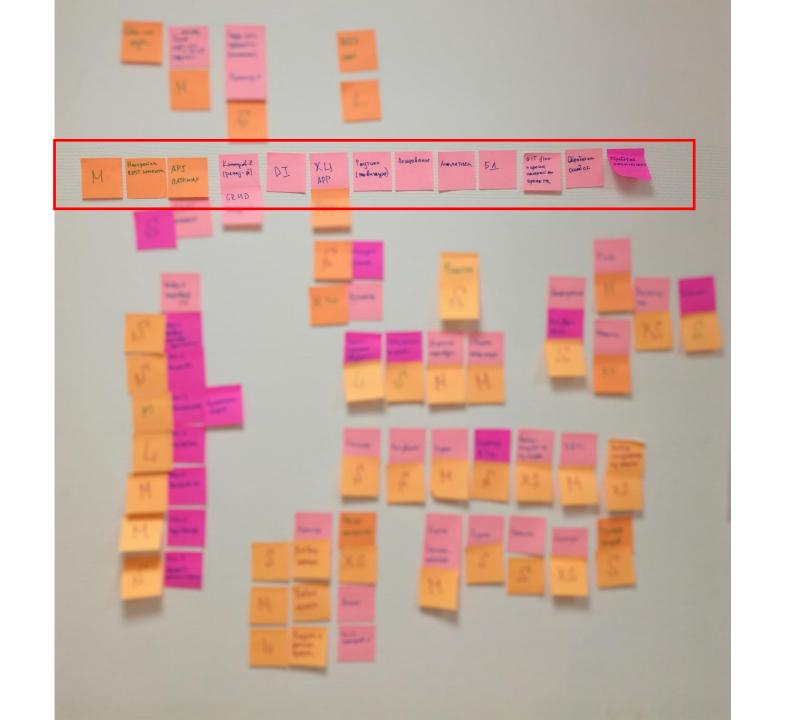


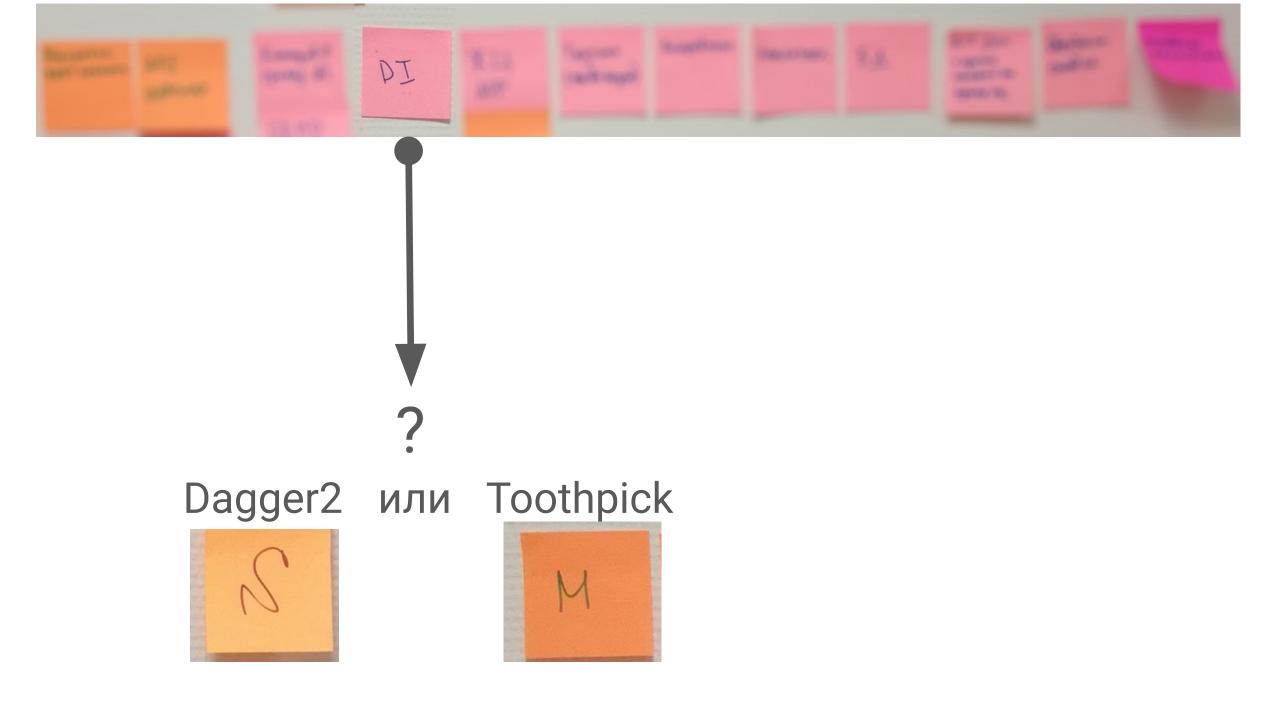
Выбираем технологии

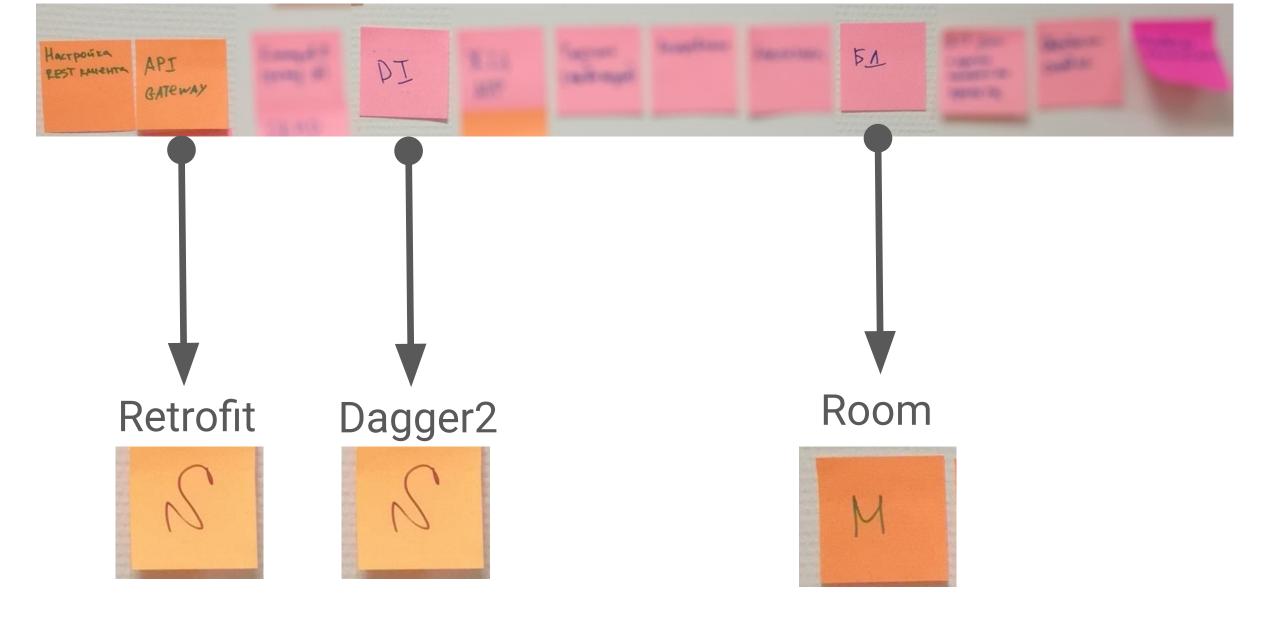


Не тратить время на изобретение велосипедов









"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
    1)
    abstract fun provideLaunchActivity(): LaunchActivity
```

@Binds @AppScope

abstract fun bindApplicationIdCreator(creator: ApplicationIdCreatorImpl): ApplicationIdCreator

@AppScope

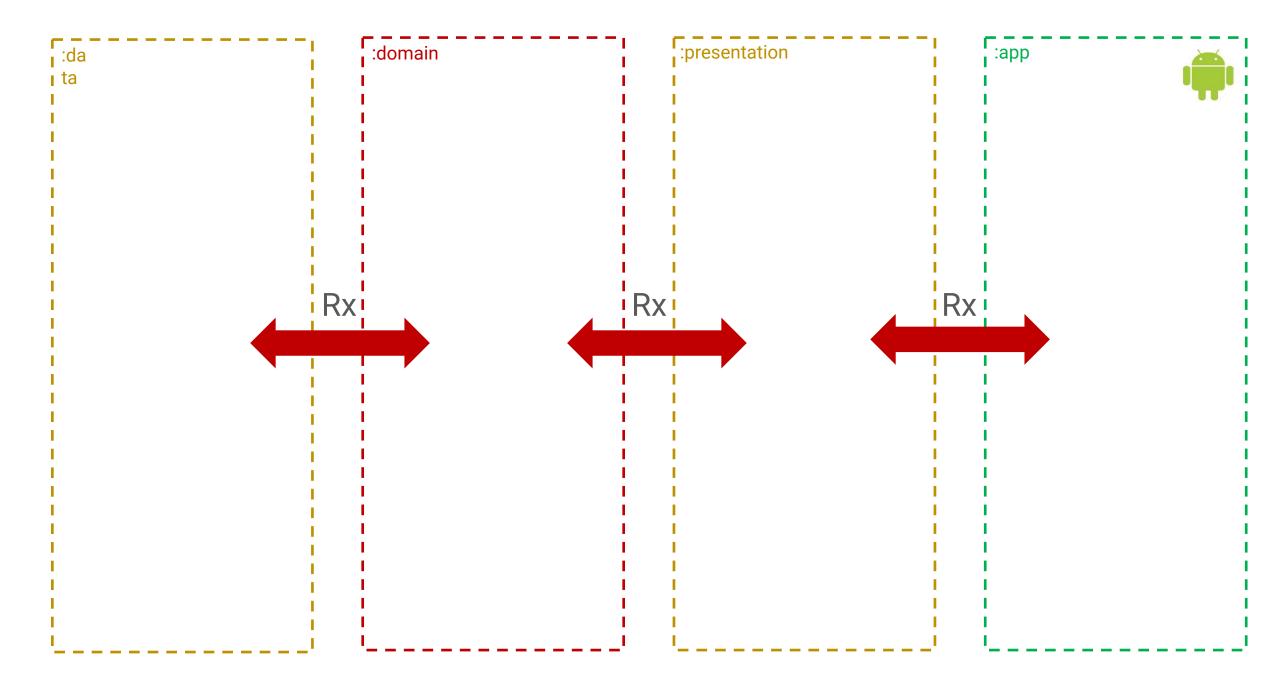
@ActivityScope

```
@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)
```

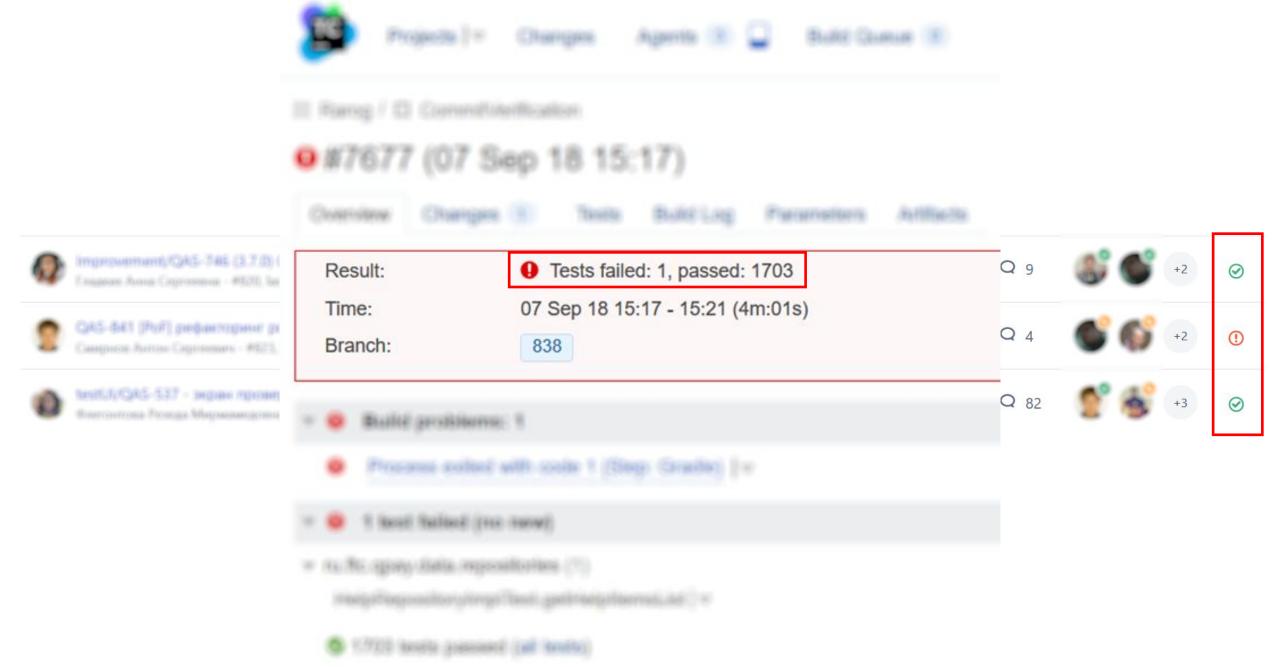
Правило

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

Тесты нам помогают



Не принимаем код без Unit-тестов

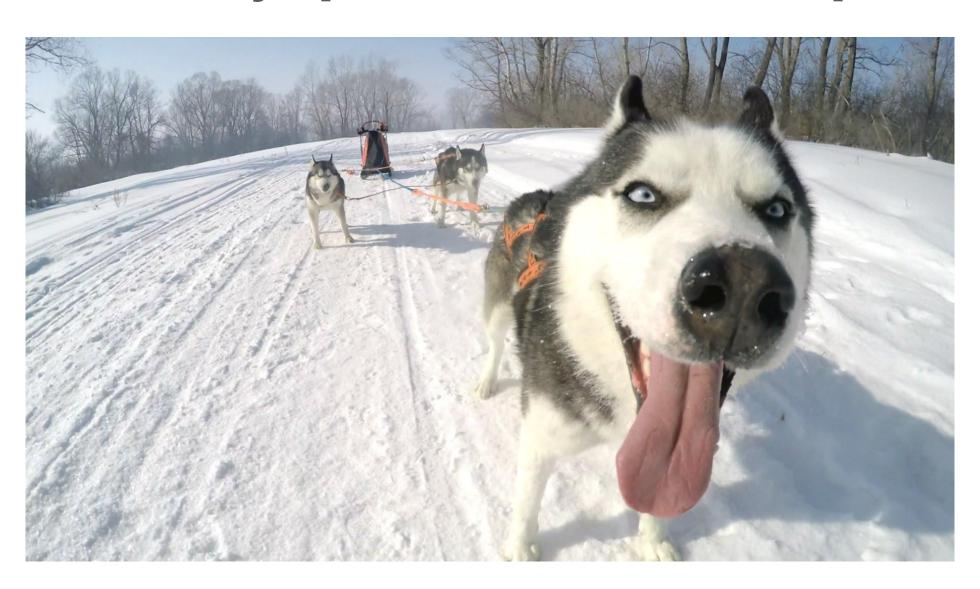


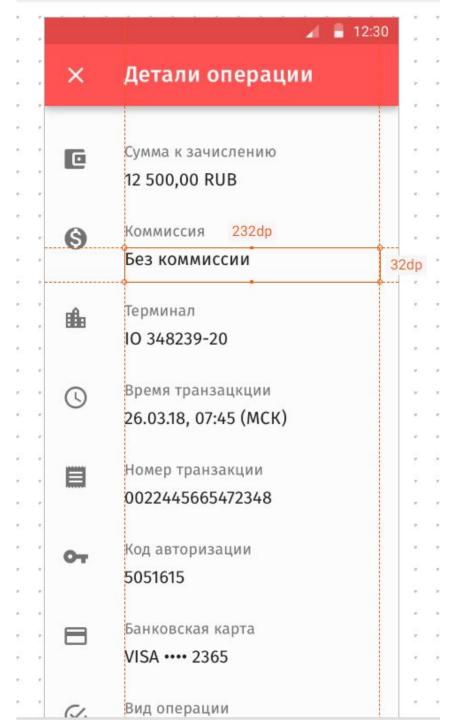




5 1741 lesis passed (all lesis)

В одной упряжке с дизайнерами









O Color Palette

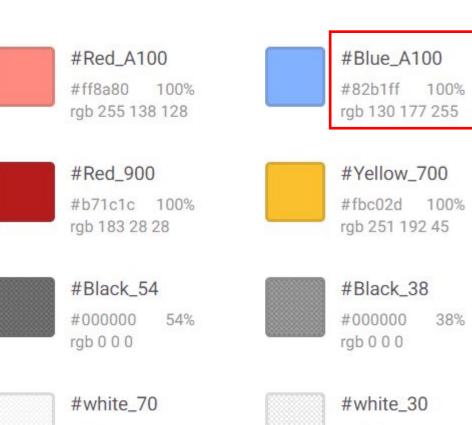


#white_100

rgb 255 255 255

100%

#ffffff



#ffffff

rgb 255 255 255

30%

#ffffff

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>
```

Colors & Text Styles Components

Aa Text Style Catalog

Display

FireSarra, Regular, 3rksp Line height: 40sp, Letter specing: 4.1

Headline

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

Tittle

FireSans, Medium, 20sp Line height 76sp, Letter specing 6.4

Subhracher

FireSams, Regular, 1/6sp Line height 3mp, Letter specing 6.1

Body 2

FireGams, Medium, 1-kep Line height 20ep, Letter specing 6.7

Buffine

FireSans, Medium, 1-kgp Line height 20sp, Letter specing ILA

Strate 1

FireSams, Regular, 1-Rep. Line height 20xo, Linter specing 6.1.

Dayter

FireSame, Regular, 1/2sp Line height Tribes Letter apseing (E.)

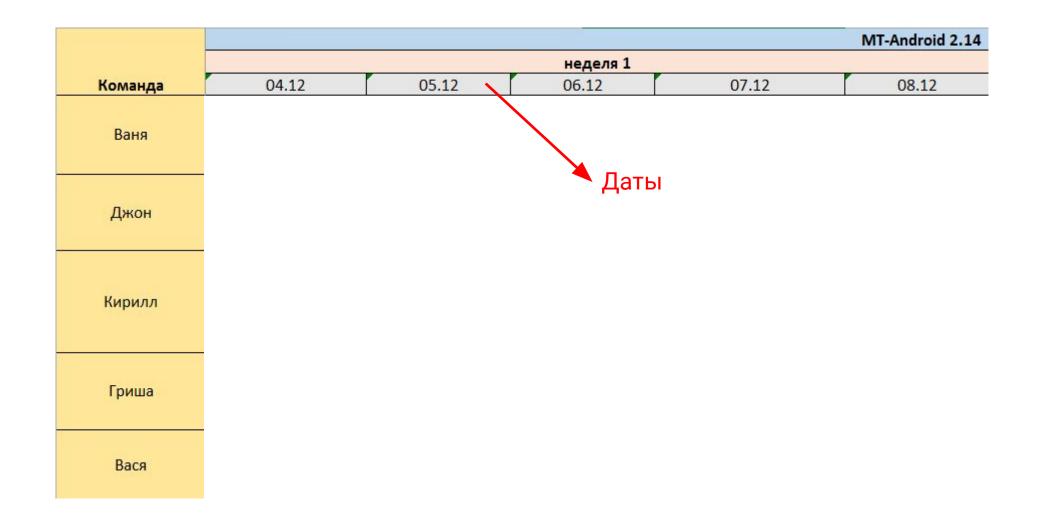
```
enurceer
style name="Blagbay">
     er androck bavickow of the college.
     android:SetSerSpecing*ofs/See
   <style name="Headline">
  <item
  name="android:textSize">24sp</item>
  <item
  name="android:letterSpacing">0</item>
  <item
  name="android:lineSpacingExtra">8sp</it
  em>
</style>
 etale reservitele">
     ne"amérold: havitélize" (2004) (/Elsaio
     on"androod: battar@acong"ob.4654/5ton
  name of androide StimeSpacingEaters*s Report it
 HEASIN MARKET Subheador" a
     na Pamainaciuli - Basatti Sicara Patagan // Elisano
     n"android: latterSpacing" 🕪 🐠 🗸
     a andreid: LineSpacingEation and appropriate
```

Не попадаем в дедлайн



Строим итеративный план спринтов

Команда
Ваня
Джон
Кирилл
Гриша
Вася

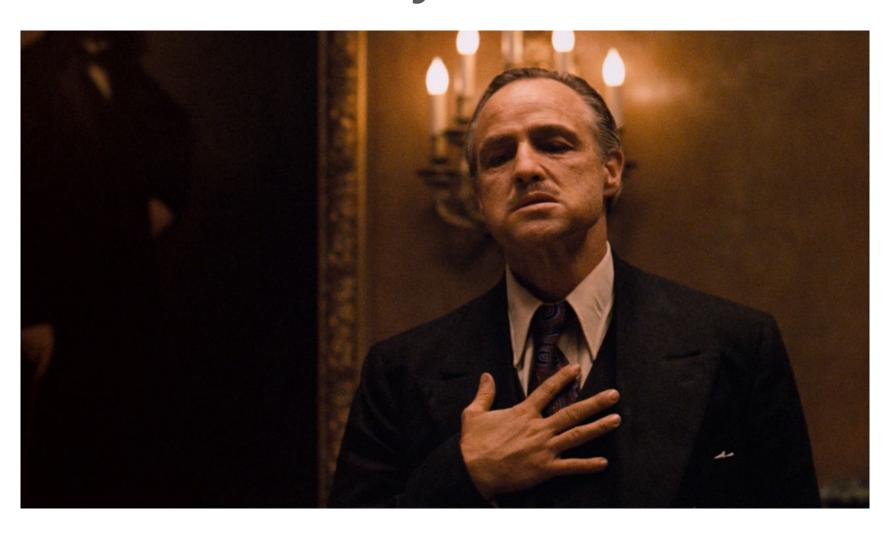


Команда					MT-Android 2.1		
	неделя 1						
	04.12	05.12	06.12	07.12	08.12		
Ваня	Новый перевод. Калькулятор - доделки			Навигация вперед	Выбор эмитент интеграция с д компонентам		
Джон							
Кирилл			Задачи				
Гриша							
Вася							

			-5		MT-Androi	d 2.1			
Команда		неделя 1							
	04.12	05.12	06.12	07.12	08.12				
Ваня	Новый пер	Новый перевод. Калькулятор - доделки			Выбор эмит интеграция компонен	с др			
Джон	Новый перевод. Получатель - domain	Новый перевод. Получатель - pr							
Кирилл				Возвраты и переход	на АРІ 2.0 в ста	эром			
Гриша	Новый перевод. Оплата - data	Новый перевод. Оплата - domain	Новый перевод. Оплата - presentation		Но	Z Z			
Вася	Новый перевод. Получатель - data	Новый перевод. Отправитель - domain	Новый перевод. Отправитель - data	Новый перевод. Отправитель - presentation	10000000	о н ра <i>м</i>			



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!

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

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

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

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

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

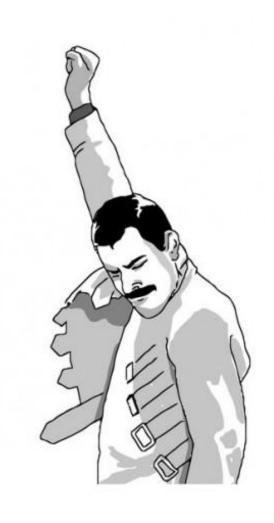
Собрать команду единомышленников и оценить масштаб

Построить архитектуру решающую задачи бизнеса

Выбрать технологии, основываясь на опыте команды

Использовать Unit-тесты, они сэкономят время на отладке

Нам это удалось, чего и вам желаем!



Спасибо 😂

Емельянов Михаил, m.emeljanov@cft.ru