KATA SOFTWARE

#Tecnología Financiera

Creating an engineering culture for better professionals and better products

Talks2Code August 2022

KATA SOFTWARE

La Suite de Kata es una solución ***SaaS*** que cuenta con componentes estándar que están listos para utilizarse en base a la configuración que el cliente requiera.

Kata **mobile**

Plataforma para Digitalizar todos los procesos y servicios que se gestionan en campo por medio de un dispositivo Móvil

Kata **bot**

Plataforma para que utiliza inteligencia artificial para flujos conversacionales

Kata **engine**

Plataforma para BackOffice y Orquestación de Productos Financieros para la aprobación y seguimiento de los créditos
Integra todos los componentes de la Kata Suite

Experiencia en el Sector Financiero

Nos enfocamos en **transformar digitalmente** y hacer más eficiente **las operaciones de nuestros clientes en el sector financiero** mediante todo el ciclo de vida del software.

BANCO PICHINCHA

Te Creemos Financiera Popular

Banco W Así de simple, así de amable.

CAME Servicios Financieros

Contactar Microfinanciera

FINANCIERA INDEPENDENCIA

mibanco

Fundación delamujer apoyamos su negocio

LIBERTAD Servicios Financieros

TSYS

Yastás.

INFONAVIT

Caja Huancayo

AFIRME

VisionFund

MICROFINANCIERA FUNDESER ¡Más cerca, más ágil!

estilos

credi Jamar

Hi Mom!

Héctor Daniel Rodríguez
Head of Engineering

/streetwizard17
@streetwizard17

Gerardo Epitacio
Software Engineer

/gerardo-epitacio
@akhha8

Engineering Culture:

“Is the set of attitudes, behaviors, beliefs, and values that impact how we build software”

Challenges we encounter

As a SaaS company, our goal is that our products never die, but they need to evolve and adapt to the market needs… and get better and better over time

Onboarding for new hires is difficult

Hiring for long term (>= 3 years)

Compensation is very competitive in the industry

Scaling knowledge is tricky

SLAs

1. People are more important than code

Companies and Teams are made of people, by the people, and for the people

Pleople will make the company achieve its goals

Genuinely care about people, and foster psychological safety

Consciuos energy is a limited resource

Measure and limit burnout

As a leader execute One-on-ones to give and get feedback, be a coach and an example for your team

2. Testing

1. There are two ways to test software
   a. Manual
   b. Automated
2. Write automated tests to validate you application's behavior faster
3. If it works, you can do it better
4. Start small, test a function, a class, a module, with time you might ended up testing 75%+ of the application's code
5. Your tests should be fast! (ms)

6.Don't tests code that you don't own

7.Use ***tests doubles***

8.Getting fancy?, check PBT.

3. Effective time management

All people on planet Earth have the same 24h, the difference is what we do with that time

Understand where the team is investing its time

Remove innecesary meetings with ***async comunication*** (email, loom)

Minimize interruptions

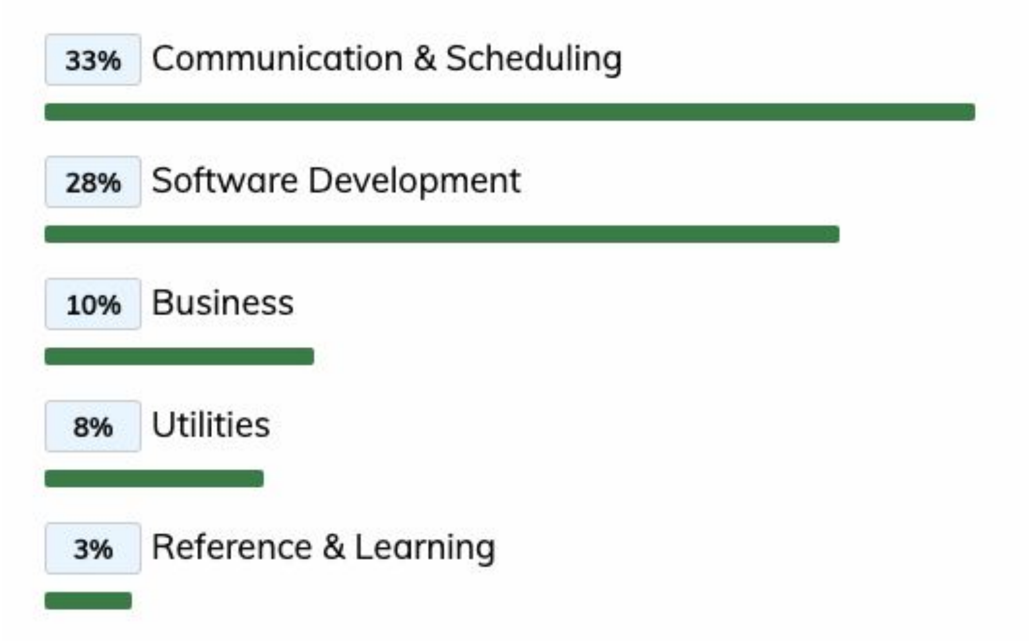
Establish time budgets for all activities

Enable the team to invest most of their time to work on the activities that will help them achieve the sprint goal

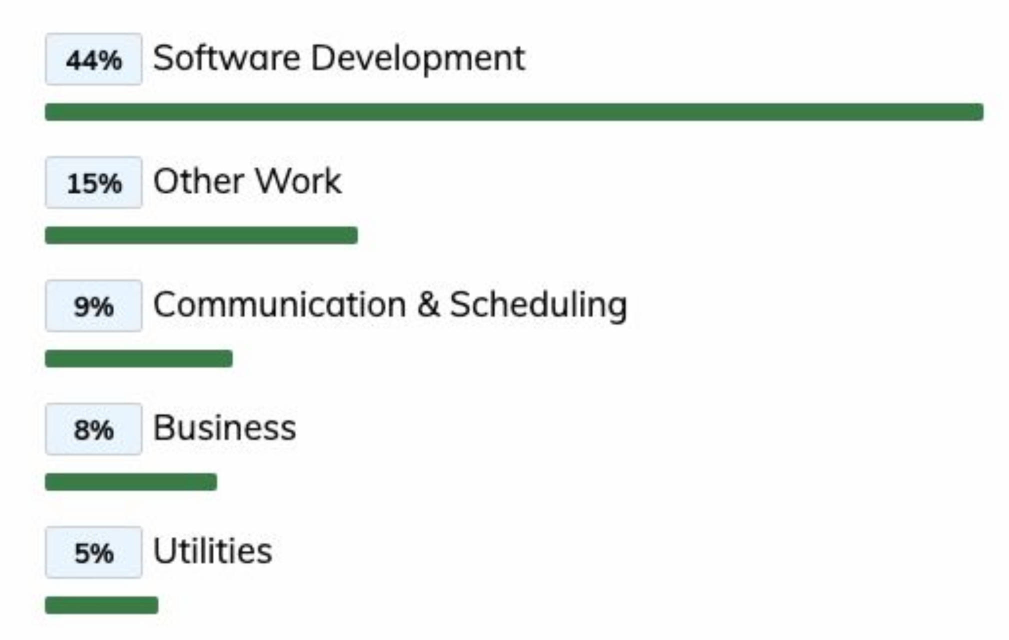
4 day week???

3. Effective time management

2021



2022



3. Effective time management

“One reason programmers dislike meetings so much is that they're on a different type of schedule from other people. Meetings cost them more.”

Maker Schedule, Managers Schedule - Paul Graham
http://www.paulgraham.com/makersschedule.html

How Future Billionaires Get Sh*t Done - Dalton Caldwell and Michael Seibel
https://youtu.be/ephzgxgOjR0

4. Measure

Numbers are far more objective than people

Do not believe what people say to you, believe the numbers they are presenting to you

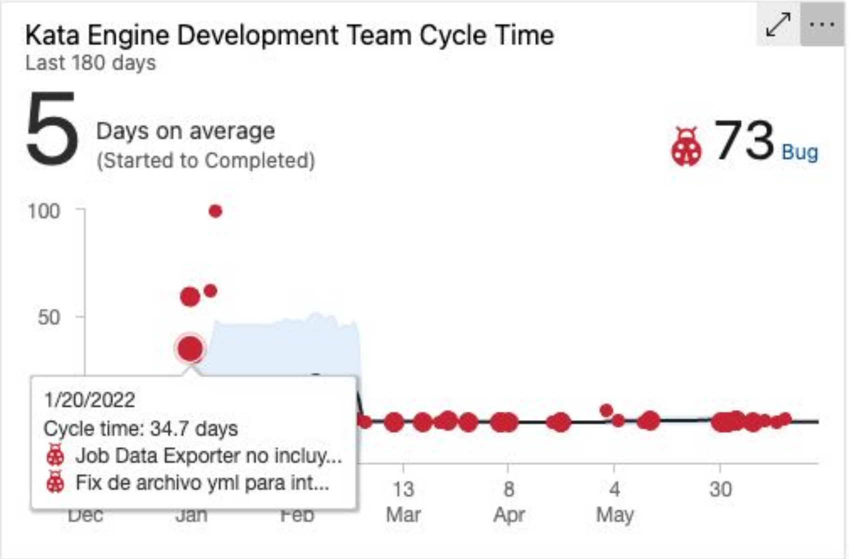
Decision making *is a process* in which you need to gather information to choose the best available option

Get data to assure if you got it right in the first place

We can’t get better in anything we don’t measure, so define and track your KPIs

- Commitment Reliability
- Time to merge
- Lead time and cycle time
- Team Velocity
- etc

KPIs 2S 2021 vs 1Q 2022



Kata Engine Development Team Cycle Time

Last 180 days

5 Days on average (Started to Completed)

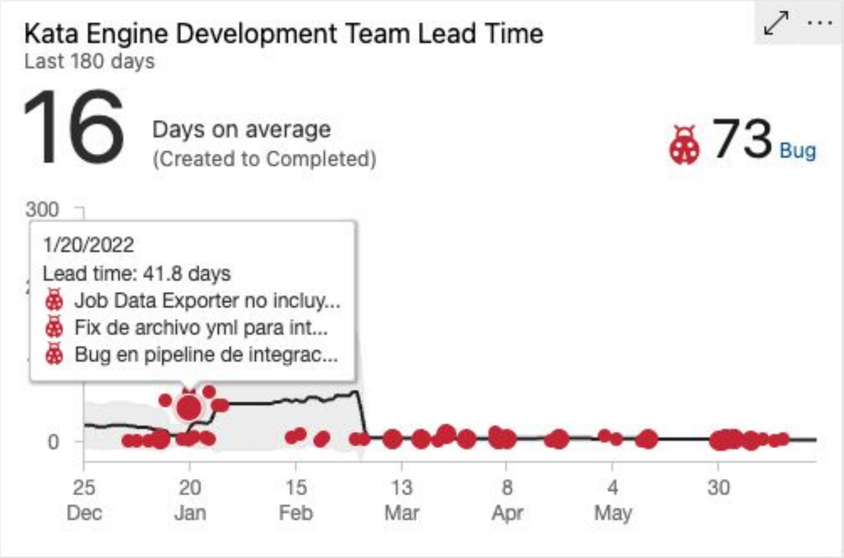
73 Bug

1/20/2022

Cycle time: 34.7 days

Job Data Exporter no incluy...

Fix de archivo yml para int...



Kata Engine Development Team Lead Time

Last 180 days

16 Days on average (Created to Completed)

73 Bug

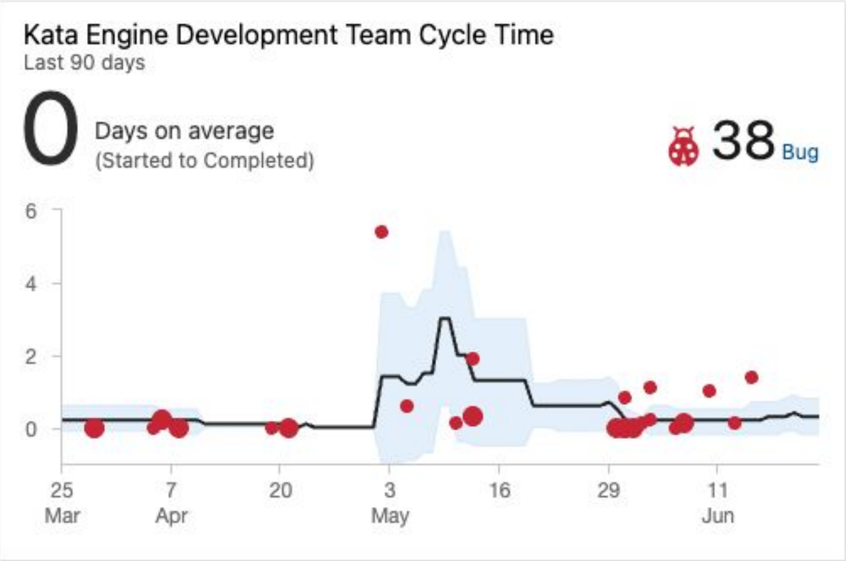
1/20/2022

Lead time: 41.8 days

Job Data Exporter no incluy...

Fix de archivo yml para int...

Bug en pipeline de integrac...

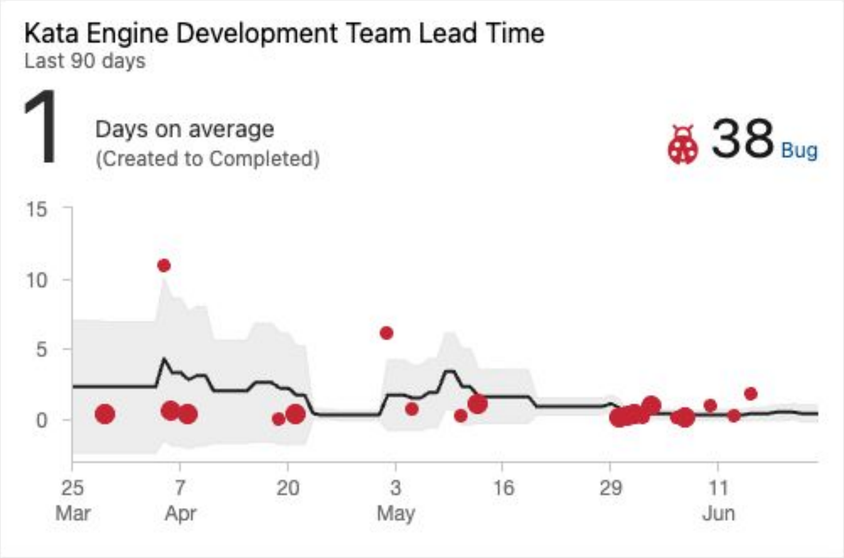


Kata Engine Development Team Cycle Time

Last 90 days

0 Days on average (Started to Completed)

38 Bug



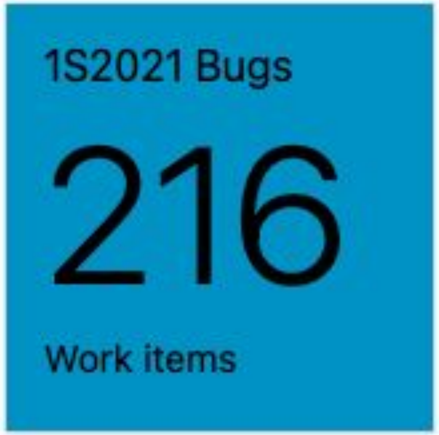
Kata Engine Development Team Lead Time

Last 90 days

1 Days on average (Created to Completed)

38 Bug

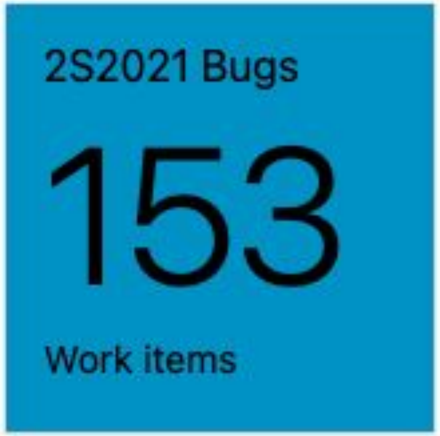
KPIs Evolution



1S2021 Bugs

216

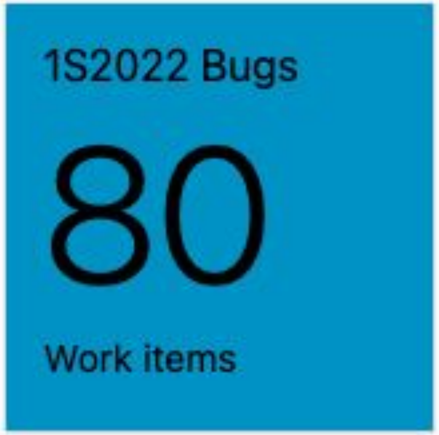
Work items



2S2021 Bugs

153

Work items



1S2022 Bugs

80

Work items

5. The team is more important than any individual

No exceptions, everyone must attach and follow the same rules

Existing rules can evolve and adapt, but we need solid arguments to change them for the whole team

No special treatment for anyone

If we achieve goals, we achieve them as a team… and if we dont, we dind’t achieve them

6. Peer Code Reviews

Its a tool for ***teamwork***

With Pull Requests you can protect your code from getting unwanted code

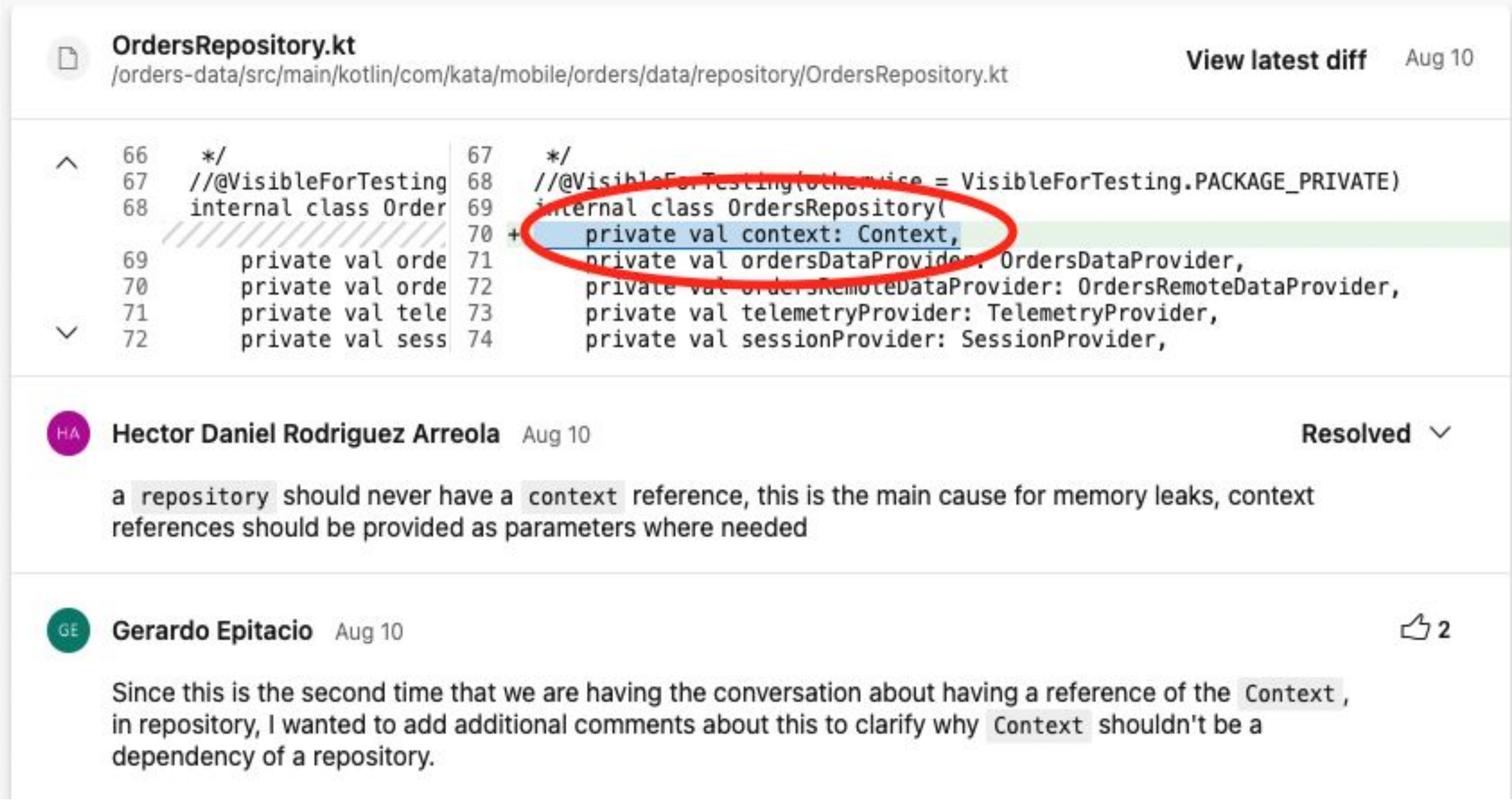
Minimize the code mistakes and its impacts

The team will learn new techniques

Help to keep code consistency

Open a discussion about technical decisions, and/or keep track about it

With Pull Requests you can protect your code from getting unwanted code



OrdersRepository.kt

/orders-data/src/main/kotlin/com/kata/mobile/orders/data/repository/OrdersRepository.kt

Aug 10

```
  */                       */
//@VisibleForTesting     //@VisibleForTesting(otherwise = VisibleForTesting.PACKAGE_PRIVATE)
internal class Order     internal class OrdersRepository(
                       +     private val context: Context,
    private val orde         private val ordersDataProvider: OrdersDataProvider,
    private val orde         private val ordersRemoteDataProvider: OrdersRemoteDataProvider,
    private val tele         private val telemetryProvider: TelemetryProvider,
    private val sess         private val sessionProvider: SessionProvider,
```

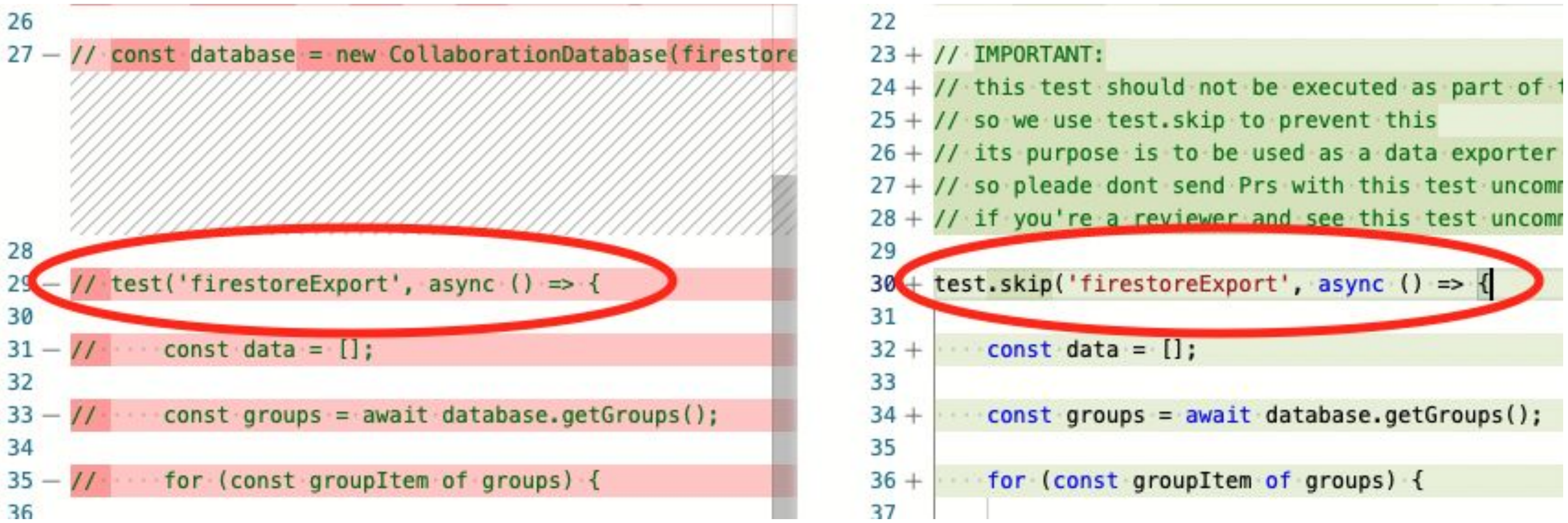
Hector Daniel Rodriguez Arreola Aug 10

a `repository` should never have a `context` reference, this is the main cause for memory leaks, context references should be provided as parameters where needed

Gerardo Epitacio Aug 10

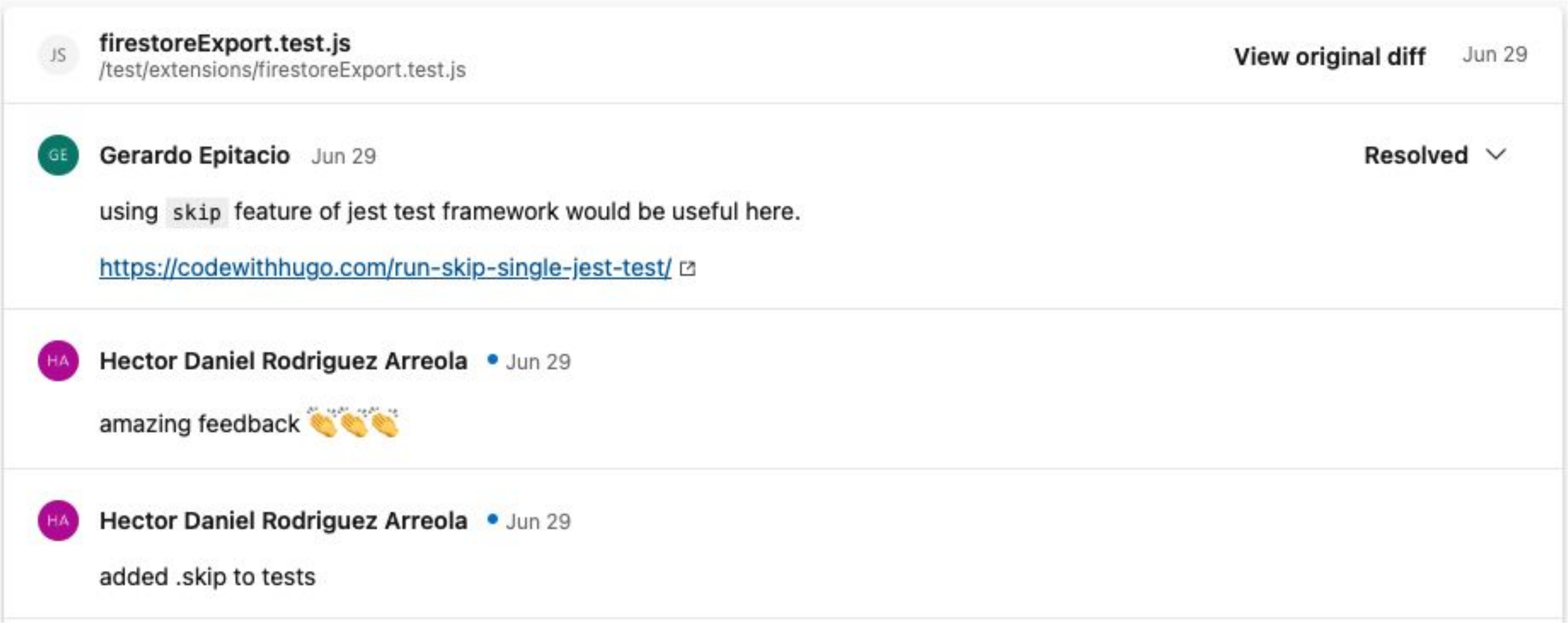
Since this is the second time that we are having the conversation about having a reference of the `Context`, in repository, I wanted to add additional comments about this to clarify why `Context` shouldn't be a dependency of a repository.

The team will learn new techniques



```
- // const database = new CollaborationDatabase(firestore

- // test('firestoreExport', async () => {

- //     const data = [];

- //     const groups = await database.getGroups();

- //     for (const groupItem of groups) {
```

```
+ // IMPORTANT:
+ // this test should not be executed as part of t
+ // so we use test.skip to prevent this
+ // its purpose is to be used as a data exporter
+ // so pleade dont send Prs with this test uncomm
+ // if you're a reviewer and see this test uncomm

+ test.skip('firestoreExport', async () => {

+     const data = [];

+     const groups = await database.getGroups();

+     for (const groupItem of groups) {
```



firestoreExport.test.js
/test/extensions/firestoreExport.test.js

Jun 29

Gerardo Epitacio Jun 29

using `skip` feature of jest test framework would be useful here.

https://codewithhugo.com/run-skip-single-jest-test/

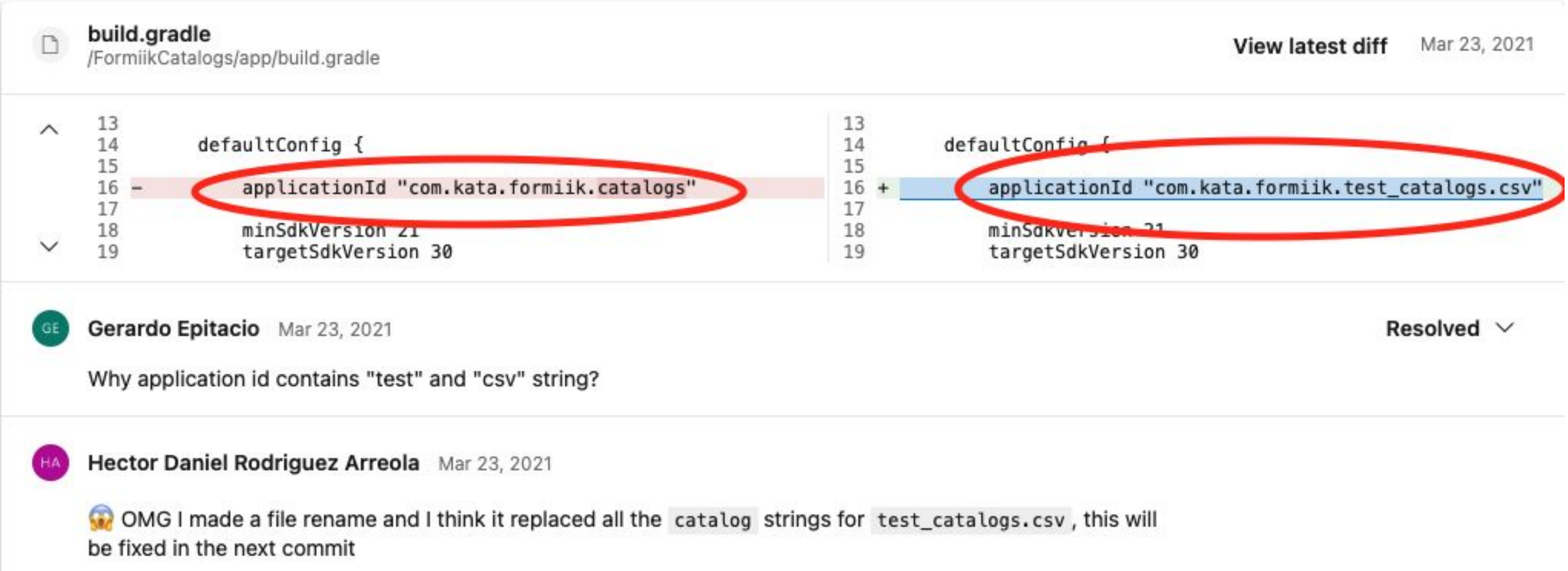
Hector Daniel Rodriguez Arreola Jun 29

amazing feedback 👏👏👏

Hector Daniel Rodriguez Arreola Jun 29

added .skip to tests

Minimize the code mistakes and its impacts



build.gradle
/FormiikCatalogs/app/build.gradle

View latest diff Mar 23, 2021

```
    defaultConfig {

-       applicationId "com.kata.formiik.catalogs"

        minSdkVersion 21
        targetSdkVersion 30
```

```
    defaultConfig {

+       applicationId "com.kata.formiik.test_catalogs.csv"

        minSdkVersion 21
        targetSdkVersion 30
```

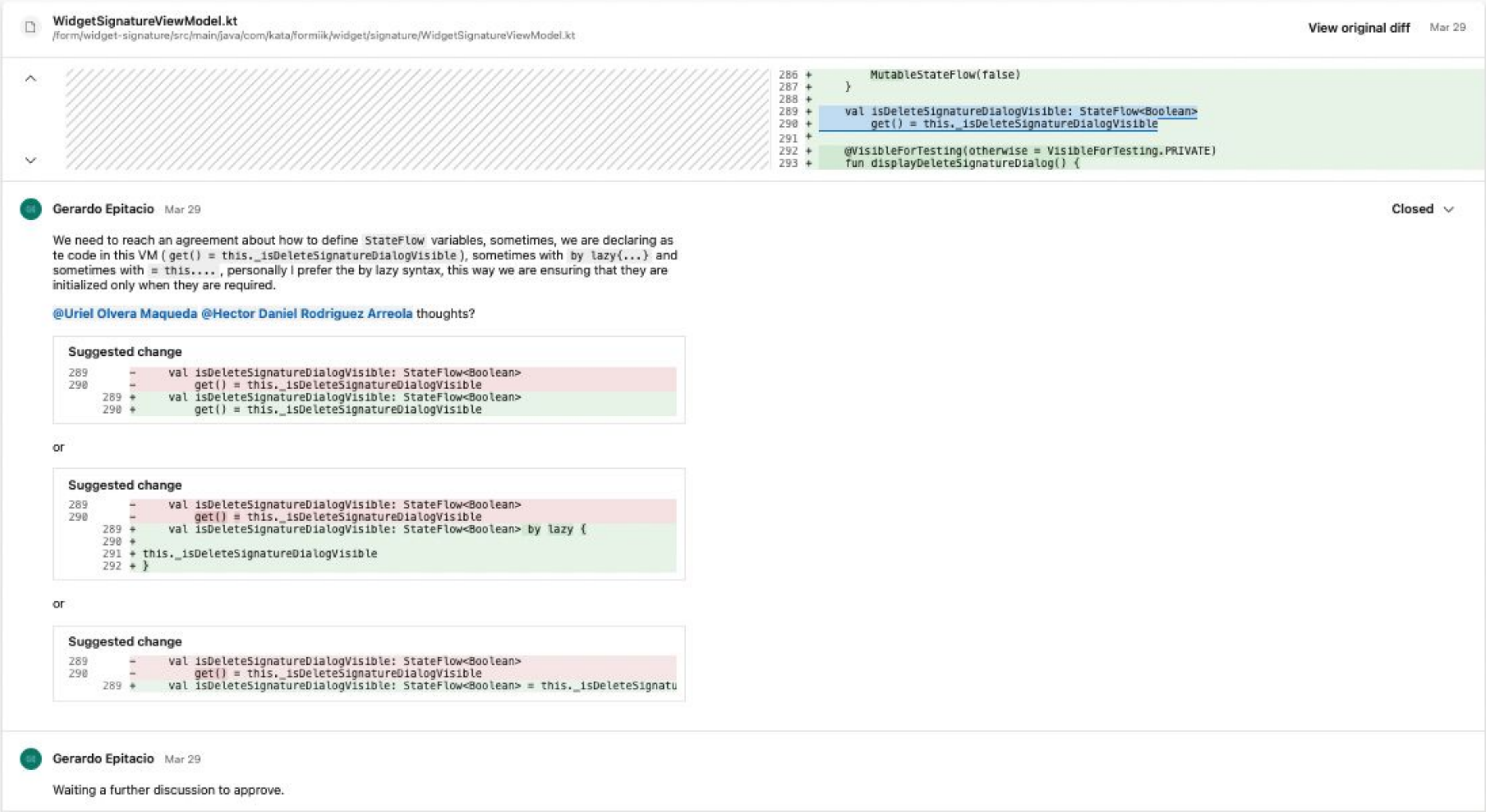
Gerardo Epitacio Mar 23, 2021 **Resolved**

Why application id contains "test" and "csv" string?

Hector Daniel Rodriguez Arreola Mar 23, 2021

😱 OMG I made a file rename and I think it replaced all the `catalog` strings for `test_catalogs.csv`, this will be fixed in the next commit

Open a discussion about technical decisions, and/or keep track about it



WidgetSignatureViewModel.kt
/form/widget-signature/src/main/java/com/kata/formiik/widget/signature/WidgetSignatureViewModel.kt

View original diff Mar 29

```
+         MutableStateFlow(false)
+     }
+
+     val isDeleteSignatureDialogVisible: StateFlow<Boolean>
+         get() = this._isDeleteSignatureDialogVisible
+
+     @VisibleForTesting(otherwise = VisibleForTesting.PRIVATE)
+     fun displayDeleteSignatureDialog() {
```

Gerardo Epitacio Mar 29

Closed

We need to reach an agreement about how to define `StateFlow` variables, sometimes, we are declaring as te code in this VM (`get() = this._isDeleteSignatureDialogVisible`), sometimes with `by lazy{...}` and sometimes with `= this....` , personally I prefer the by lazy syntax, this way we are ensuring that they are initialized only when they are required.

@Uriel Olvera Maqueda @Hector Daniel Rodriguez Arreola thoughts?

Suggested change

```
-     val isDeleteSignatureDialogVisible: StateFlow<Boolean>
-         get() = this._isDeleteSignatureDialogVisible
+     val isDeleteSignatureDialogVisible: StateFlow<Boolean>
+         get() = this._isDeleteSignatureDialogVisible
```

or

Suggested change

```
-     val isDeleteSignatureDialogVisible: StateFlow<Boolean>
-         get() = this._isDeleteSignatureDialogVisible
+     val isDeleteSignatureDialogVisible: StateFlow<Boolean> by lazy {
+
+ this._isDeleteSignatureDialogVisible
+ }
```

or

Suggested change

```
-     val isDeleteSignatureDialogVisible: StateFlow<Boolean>
-         get() = this._isDeleteSignatureDialogVisible
+     val isDeleteSignatureDialogVisible: StateFlow<Boolean> = this._isDeleteSignatu
```

Gerardo Epitacio Mar 29

Waiting a further discussion to approve.

7. Code Consistency

“All code in the team must seem to have been written by the same person”

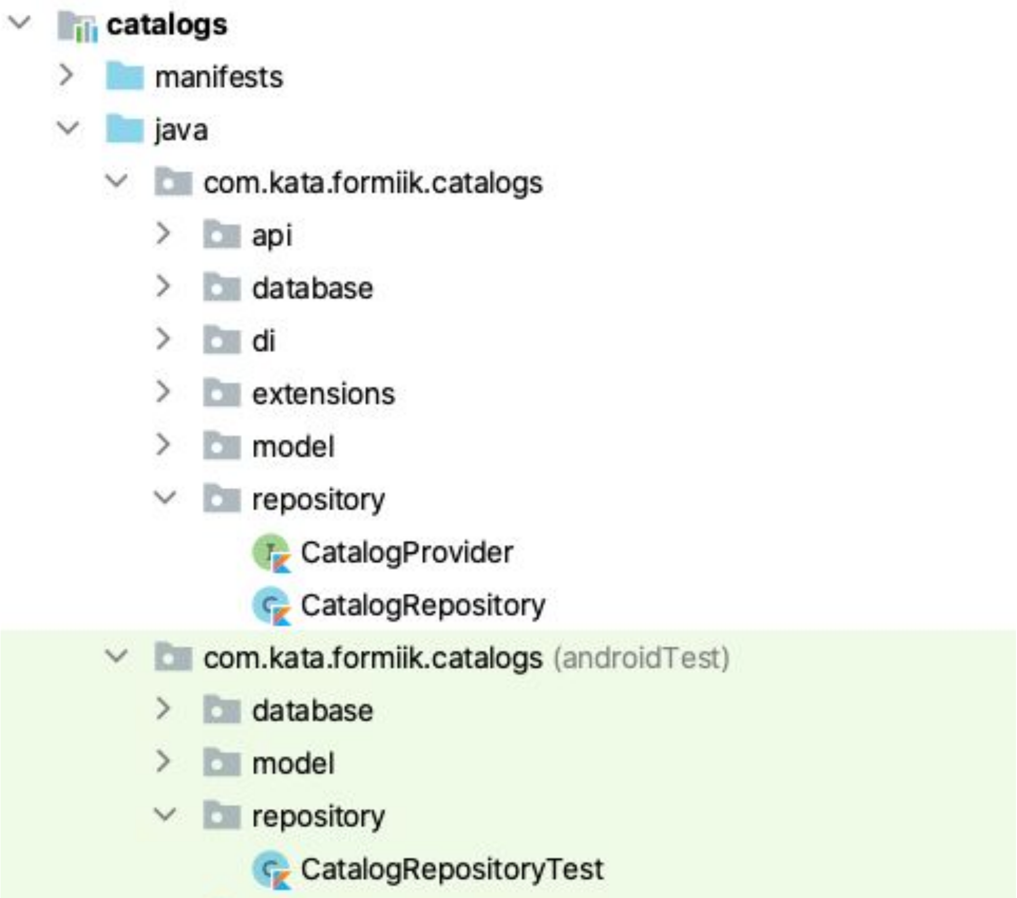
The team focuses more on building new features, and less in understanding the existing code base

Reduce the learning curve for new hires

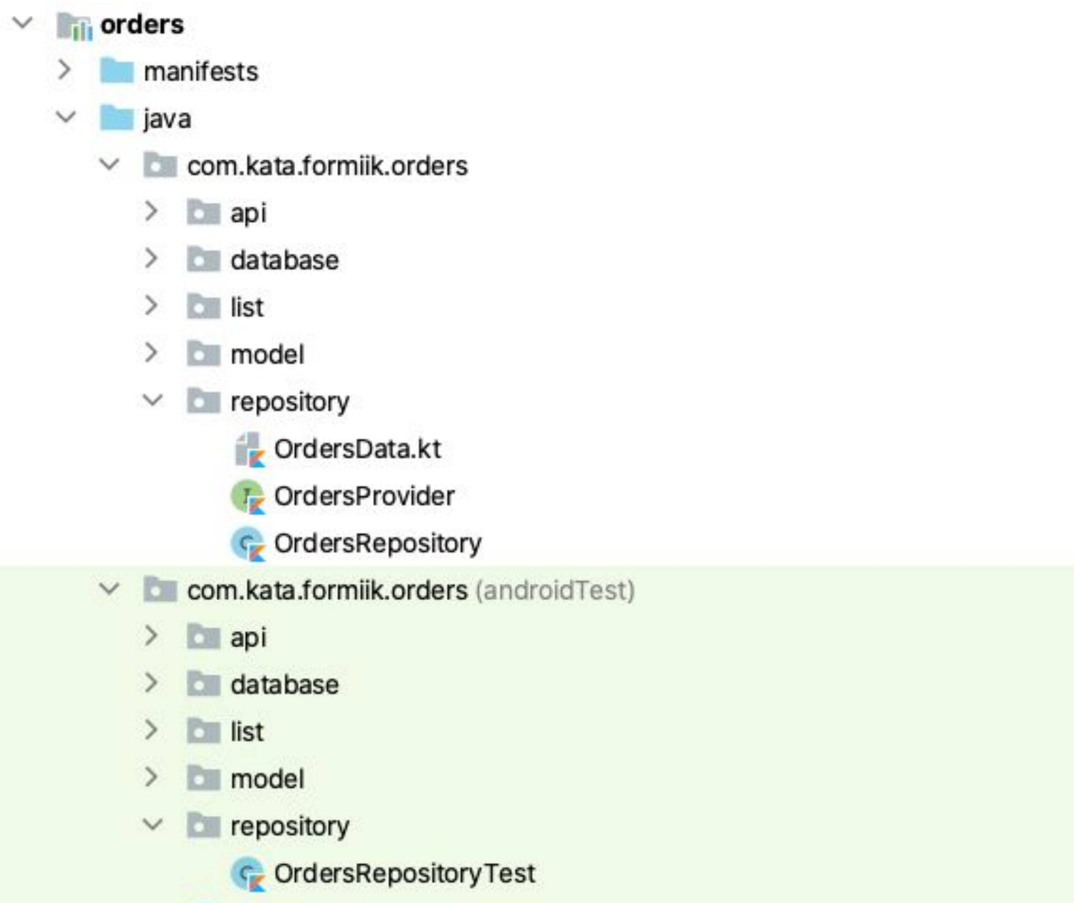
Tool adoption and implementation for developer productivity becomes easier

Evolve your product faster

Project structure and file naming

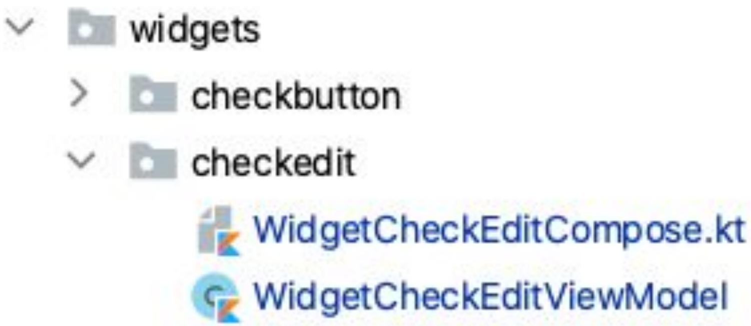


```
catalogs
  manifests
  java
    com.kata.formiik.catalogs
      api
      database
      di
      extensions
      model
      repository
        CatalogProvider
        CatalogRepository
    com.kata.formiik.catalogs (androidTest)
      database
      model
      repository
        CatalogRepositoryTest
```

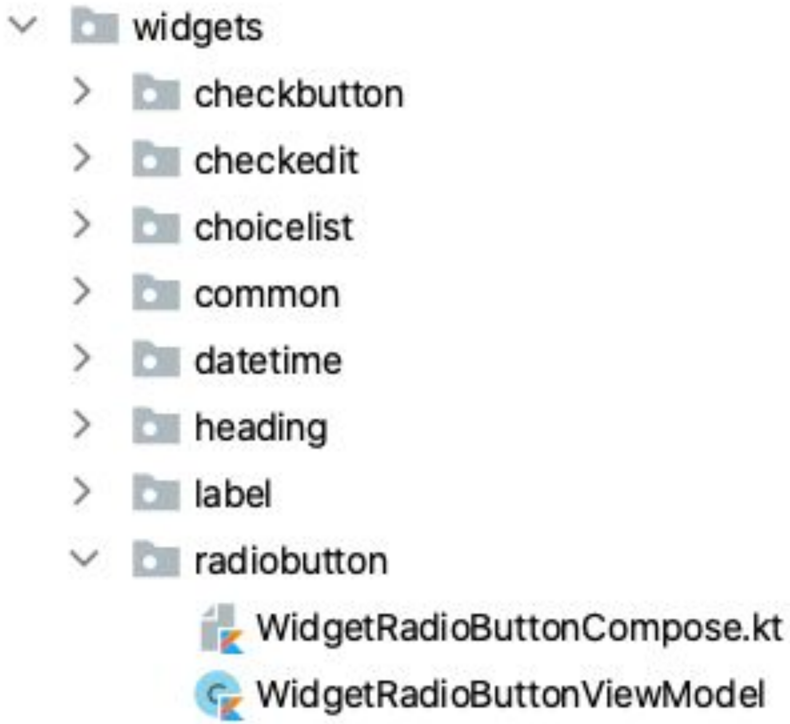


```
orders
  manifests
  java
    com.kata.formiik.orders
      api
      database
      list
      model
      repository
        OrdersData.kt
        OrdersProvider
        OrdersRepository
    com.kata.formiik.orders (androidTest)
      api
      database
      list
      model
      repository
        OrdersRepositoryTest
```

Class names and separation of concerns

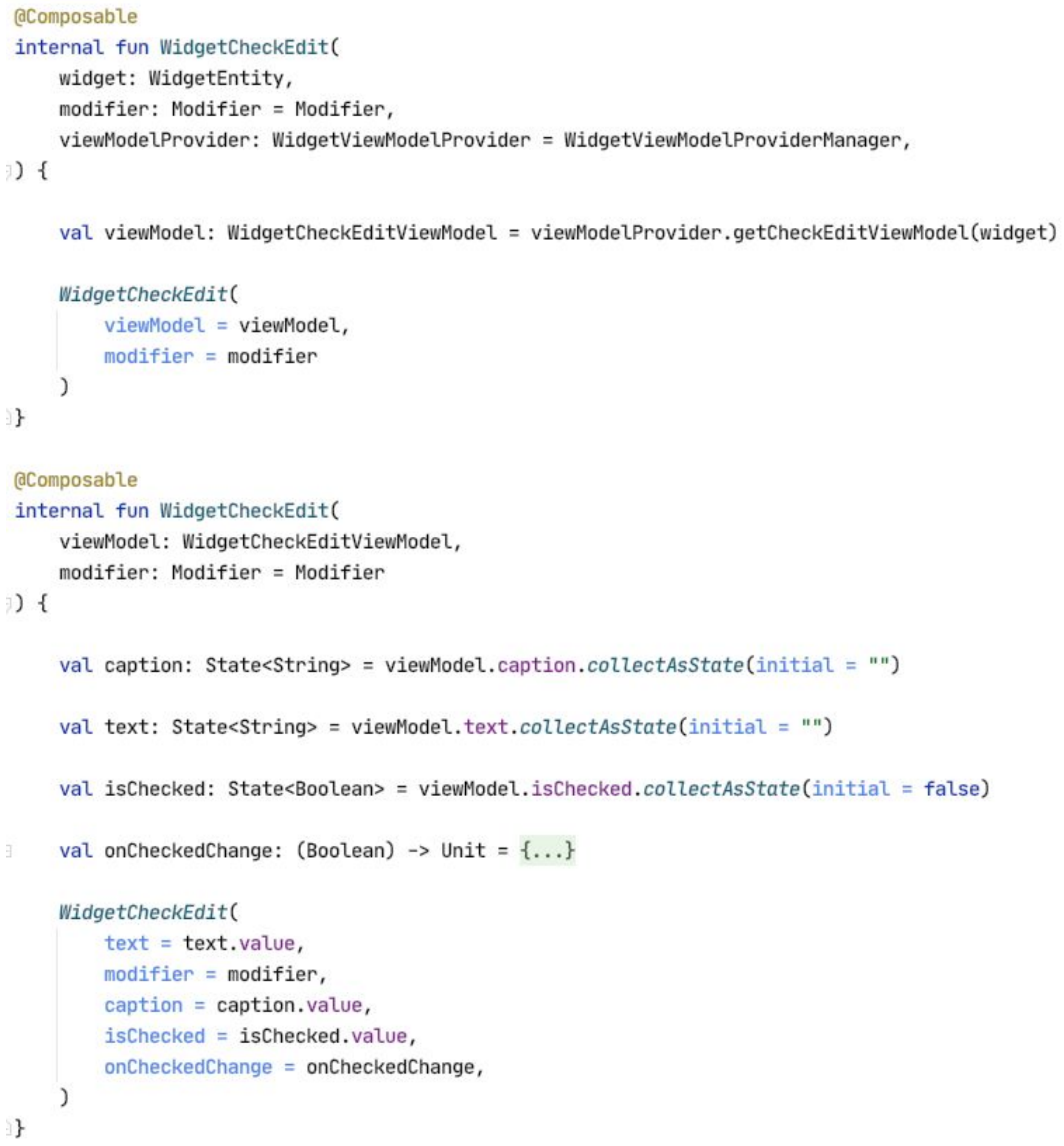


- widgets
  - checkbutton
  - checkedit
    - WidgetCheckEditCompose.kt
    - WidgetCheckEditViewModel

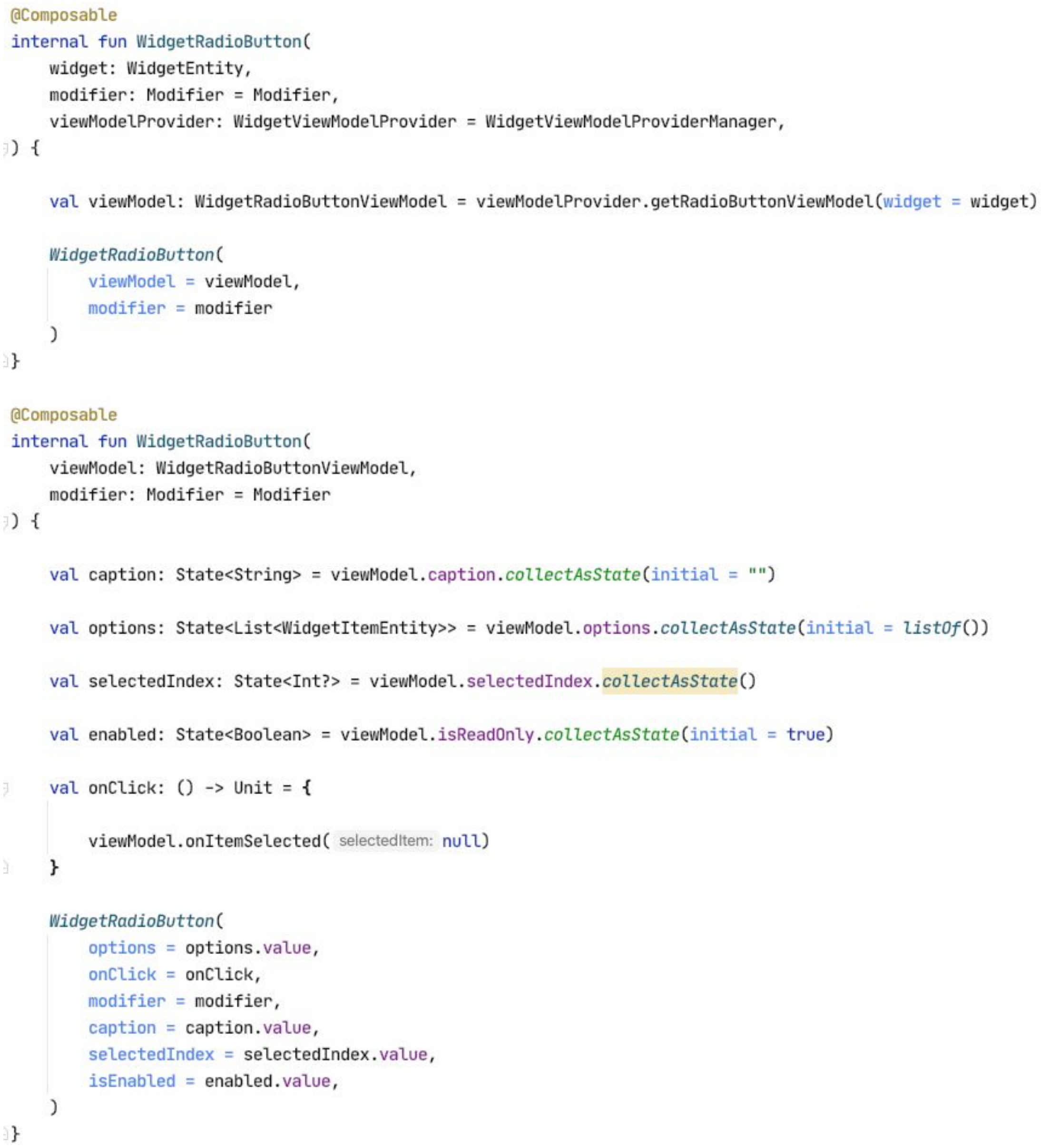


- widgets
  - checkbutton
  - checkedit
  - choicelist
  - common
  - datetime
  - heading
  - label
  - radiobutton
    - WidgetRadioButtonCompose.kt
    - WidgetRadioButtonViewModel

Function implementation and parametrization



```kotlin
@Composable
internal fun WidgetCheckEdit(
    widget: WidgetEntity,
    modifier: Modifier = Modifier,
    viewModelProvider: WidgetViewModelProvider = WidgetViewModelProviderManager,
) {

    val viewModel: WidgetCheckEditViewModel = viewModelProvider.getCheckEditViewModel(widget)

    WidgetCheckEdit(
        viewModel = viewModel,
        modifier = modifier
    )
}

@Composable
internal fun WidgetCheckEdit(
    viewModel: WidgetCheckEditViewModel,
    modifier: Modifier = Modifier
) {

    val caption: State<String> = viewModel.caption.collectAsState(initial = "")

    val text: State<String> = viewModel.text.collectAsState(initial = "")

    val isChecked: State<Boolean> = viewModel.isChecked.collectAsState(initial = false)

    val onCheckedChange: (Boolean) -> Unit = {...}

    WidgetCheckEdit(
        text = text.value,
        modifier = modifier,
        caption = caption.value,
        isChecked = isChecked.value,
        onCheckedChange = onCheckedChange,
    )
}
```



```kotlin
@Composable
internal fun WidgetRadioButton(
    widget: WidgetEntity,
    modifier: Modifier = Modifier,
    viewModelProvider: WidgetViewModelProvider = WidgetViewModelProviderManager,
) {

    val viewModel: WidgetRadioButtonViewModel = viewModelProvider.getRadioButtonViewModel(widget = widget)

    WidgetRadioButton(
        viewModel = viewModel,
        modifier = modifier
    )
}

@Composable
internal fun WidgetRadioButton(
    viewModel: WidgetRadioButtonViewModel,
    modifier: Modifier = Modifier
) {

    val caption: State<String> = viewModel.caption.collectAsState(initial = "")

    val options: State<List<WidgetItemEntity>> = viewModel.options.collectAsState(initial = listOf())

    val selectedIndex: State<Int?> = viewModel.selectedIndex.collectAsState()

    val enabled: State<Boolean> = viewModel.isReadOnly.collectAsState(initial = true)

    val onClick: () -> Unit = {

        viewModel.onItemSelected( selectedItem: null)
    }

    WidgetRadioButton(
        options = options.value,
        onClick = onClick,
        modifier = modifier,
        caption = caption.value,
        selectedIndex = selectedIndex.value,
        isEnabled = enabled.value,
    )
}
```

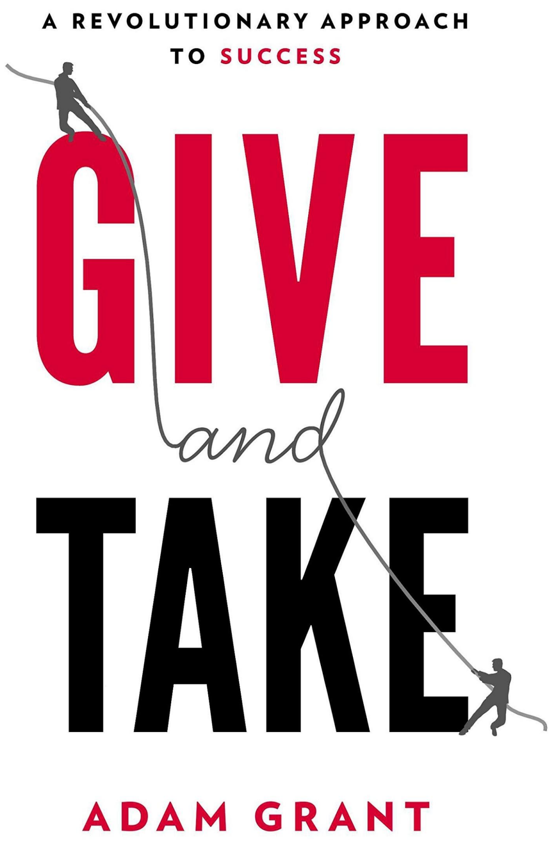
8. Be better everyday.. and make better the ones around you.

Be a ***giver,*** and protect other givers

Invest on people because they will help you to achieve your goals

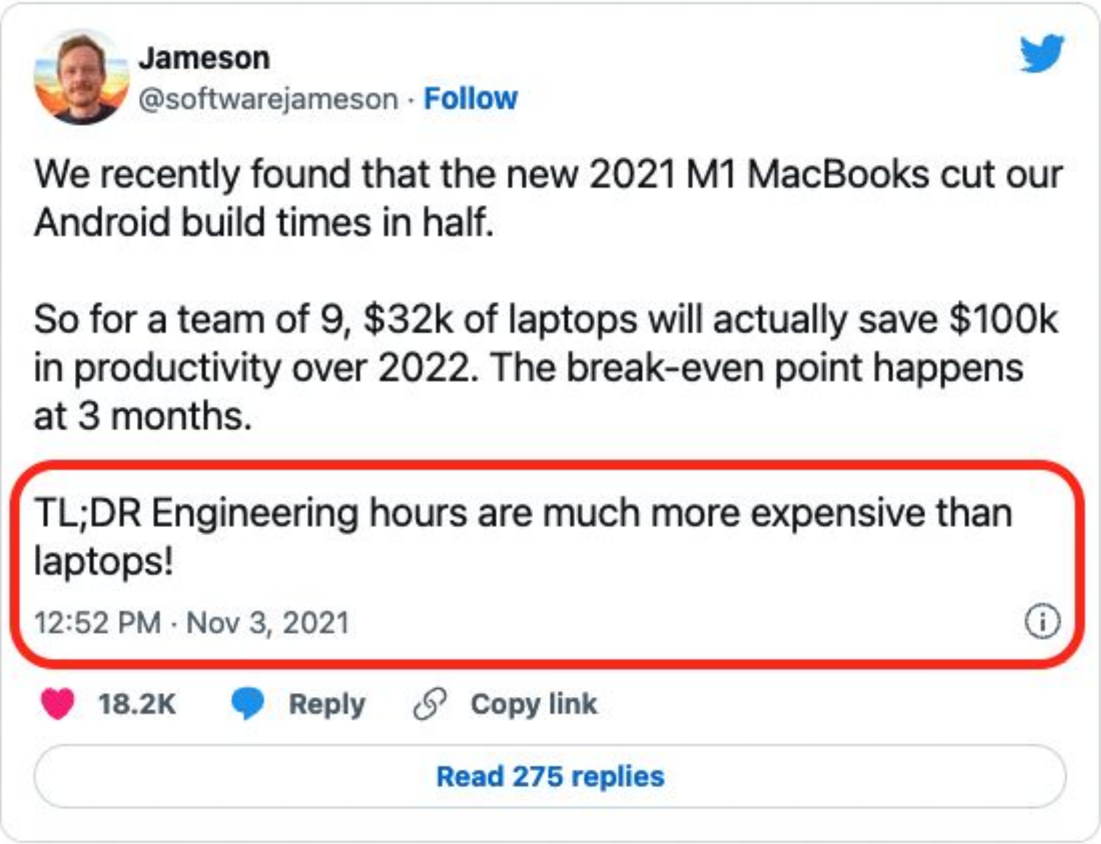
Do not create dependencies on the individuals

Distribute the knowledge in all team members



9. Developer productivity

Give them tools to build fast



Jameson
@softwarejameson · Follow

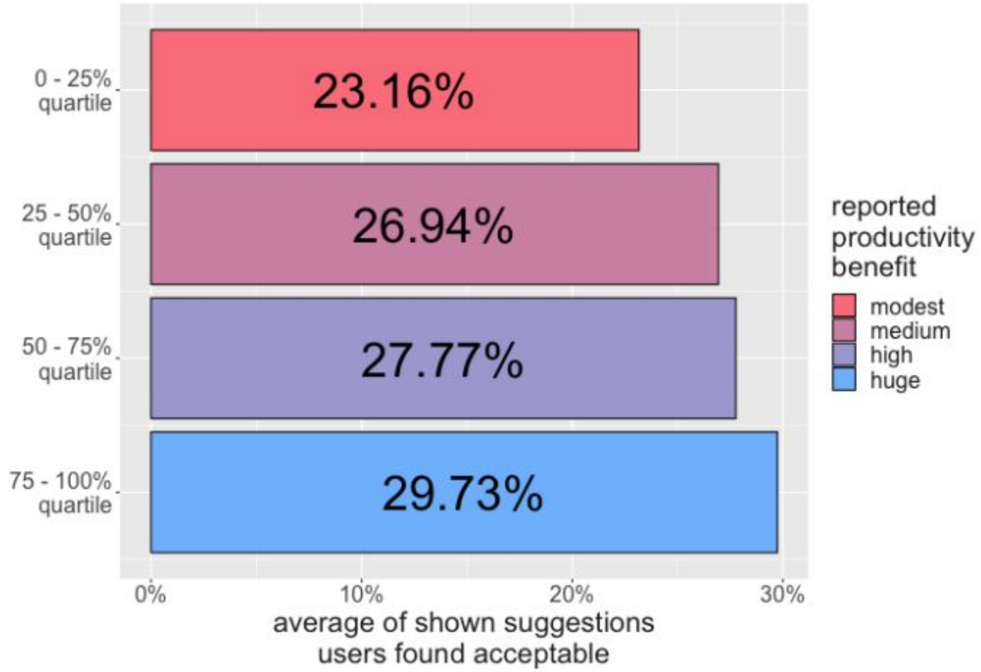
We recently found that the new 2021 M1 MacBooks cut our Android build times in half.

So for a team of 9, $32k of laptops will actually save $100k in productivity over 2022. The break-even point happens at 3 months.

TL;DR Engineering hours are much more expensive than laptops!

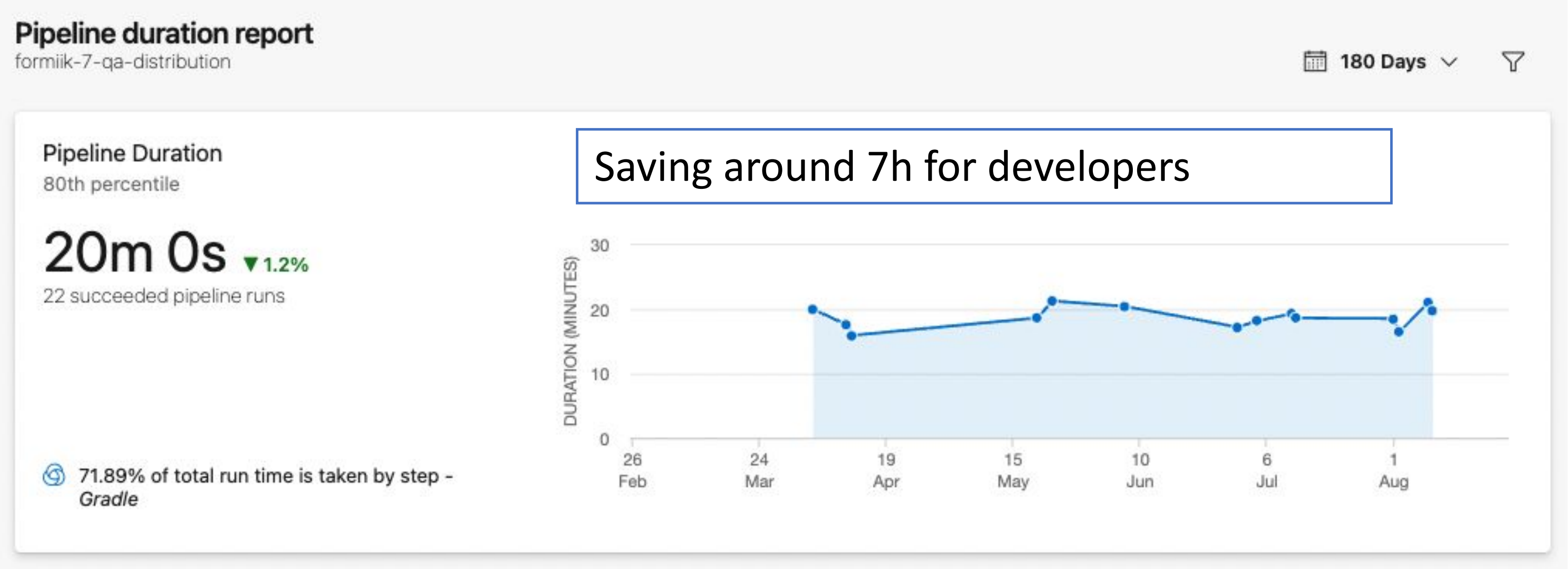
12:52 PM · Nov 3, 2021

Developers who report the highest productivity gains with GitHub Copilot also accept the largest number of shown code suggestions



9. Developer productivity

Automate as most as you can – CI/CD pipelines



Pipeline duration report
formiik-7-qa-distribution

180 Days

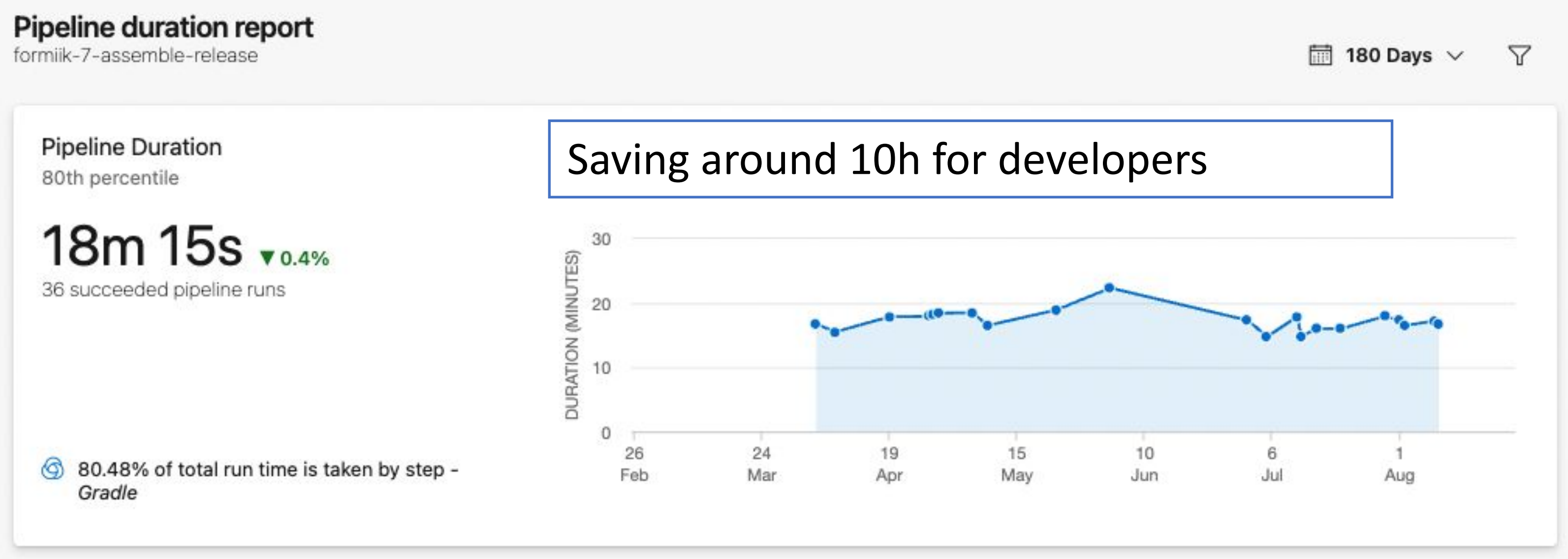
Pipeline Duration
80th percentile

20m 0s ▼1.2%

22 succeeded pipeline runs

71.89% of total run time is taken by step - *Gradle*

Saving around 7h for developers



Pipeline duration report
formiik-7-assemble-release

180 Days

Pipeline Duration
80th percentile

18m 15s ▼0.4%

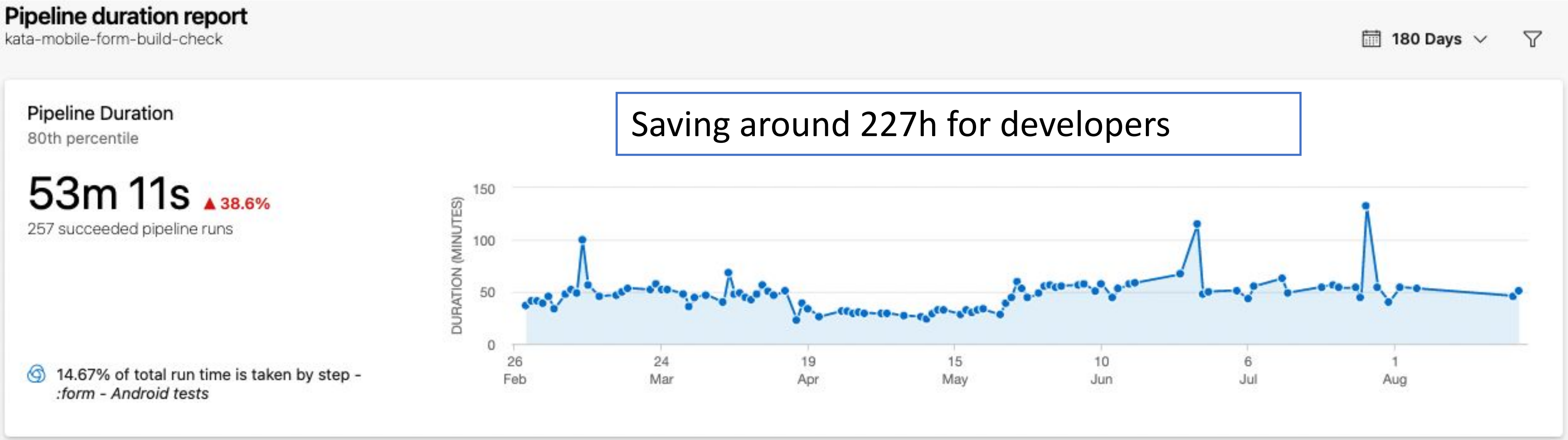
36 succeeded pipeline runs

80.48% of total run time is taken by step - *Gradle*

Saving around 10h for developers

9. Developer productivity

Automate as most as you can – test check & reporting



Pipeline duration report

kata-mobile-form-build-check

180 Days

Pipeline Duration

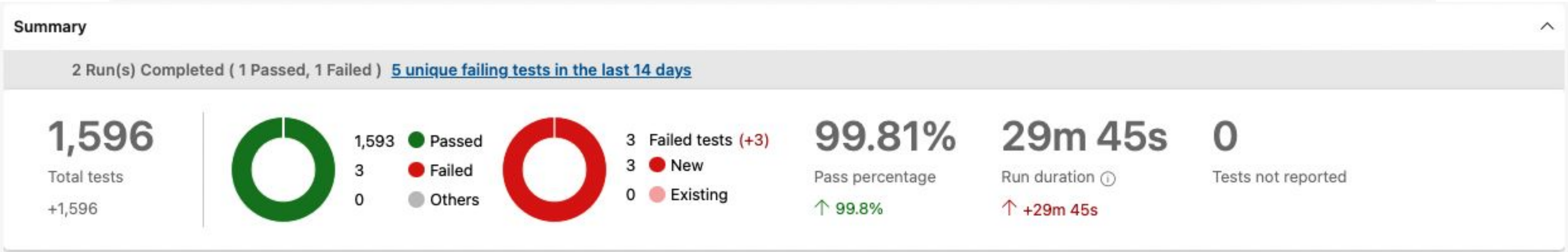
80th percentile

53m 11s ▲ 38.6%

257 succeeded pipeline runs

14.67% of total run time is taken by step - *:form - Android tests*

Saving around 227h for developers



Summary

2 Run(s) Completed (1 Passed, 1 Failed) 5 unique failing tests in the last 14 days

1,596

Total tests

+1,596

1,593 ● Passed

3 ● Failed

0 ● Others

3 Failed tests (+3)

3 ● New

0 ● Existing

99.81%

Pass percentage

↑ 99.8%

29m 45s

Run duration ⓘ

↑ +29m 45s

0

Tests not reported

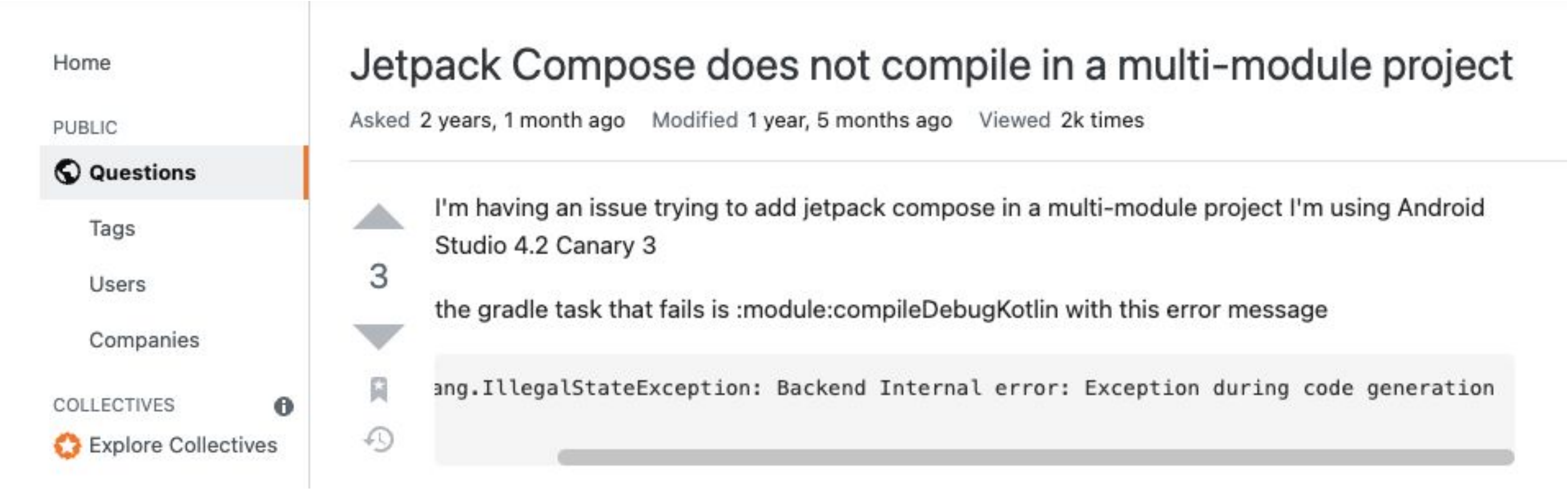
10. Innovate and Experiment

API-First
- Testing and Development at the same time

Android
- Jetpack Compose
- Android Instant Apps

Authentication
- Passwordless biometric authentication

Testing
- Property based testing



Jetpack Compose does not compile in a multi-module project

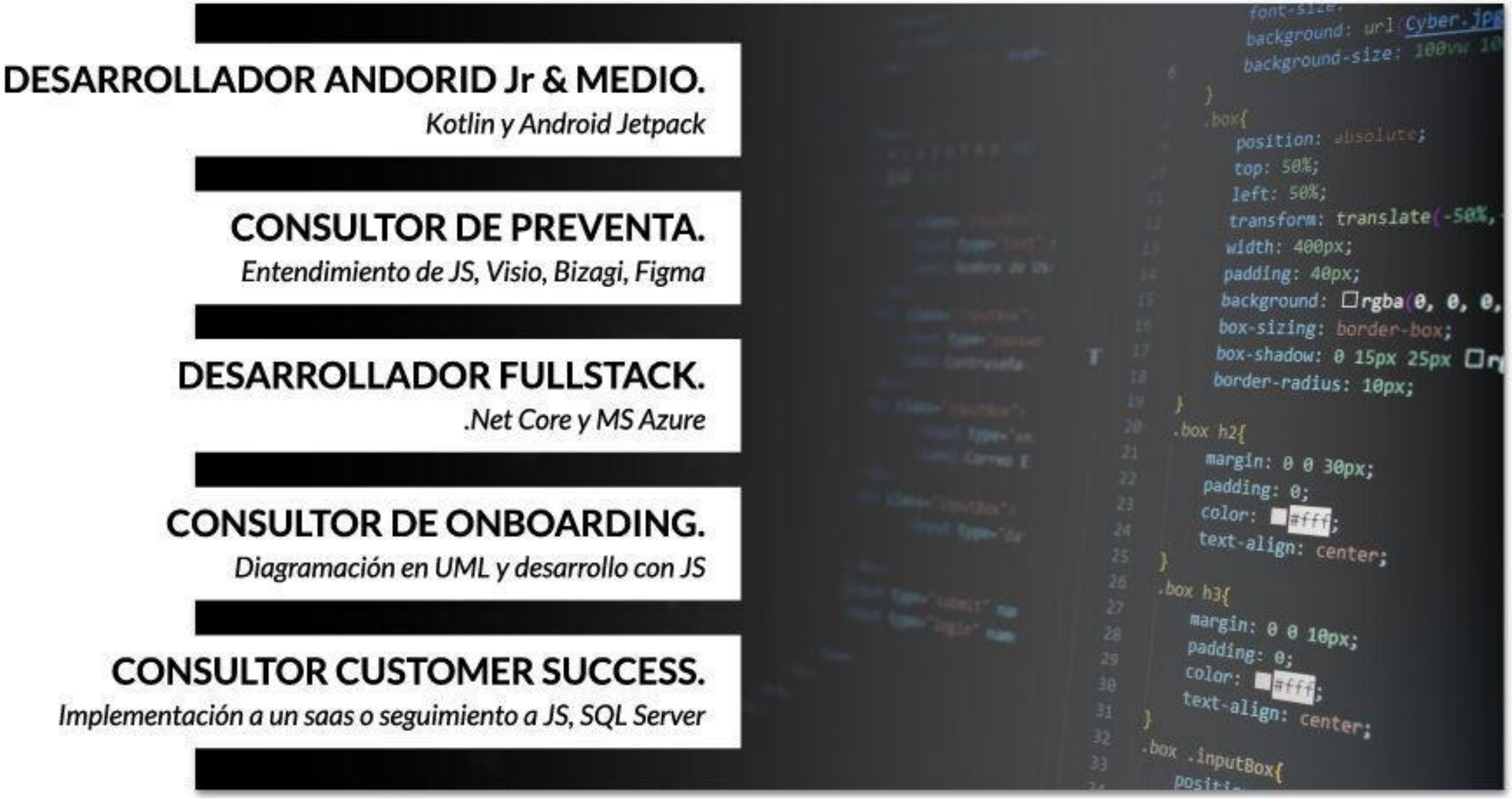
Asked 2 years, 1 month ago Modified 1 year, 5 months ago Viewed 2k times

I'm having an issue trying to add jetpack compose in a multi-module project I'm using Android Studio 4.2 Canary 3

the gradle task that fails is :module:compileDebugKotlin with this error message

```
ang.IllegalStateException: Backend Internal error: Exception during code generation
```

We are Hiring!



DESARROLLADOR ANDORID Jr & MEDIO.
Kotlin y Android Jetpack

CONSULTOR DE PREVENTA.
Entendimiento de JS, Visio, Bizagi, Figma

DESARROLLADOR FULLSTACK.
.Net Core y MS Azure

CONSULTOR DE ONBOARDING.
Diagramación en UML y desarrollo con JS

CONSULTOR CUSTOMER SUCCESS.
Implementación a un saas o seguimiento a JS, SQL Server

```
    background: url(Cyber.JPG
    background-size: 100vw 10
}
.box{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-
    width: 400px;
    padding: 40px;
    background: rgba(0, 0, 0,
    box-sizing: border-box;
    box-shadow: 0 15px 25px r
    border-radius: 10px;
}
.box h2{
    margin: 0 0 30px;
    padding: 0;
    color: #fff;
    text-align: center;
}
.box h3{
    margin: 0 0 10px;
    padding: 0;
    color: #fff;
    text-align: center;
}
.box .inputBox{
```

linkedin.com/company/kata-software/jobs/

https://bit.ly/3pGPn7C

We invite you to our Kata Sessions!

Junio 2022 - Seeing The Matrix by testing everything with code
Best Moments: https://vimeo.com/725055793
Full Session: https://vimeo.com/725038669

Septiembre 2022 - ???

linkedin.com/company/kata-software

Thanks!