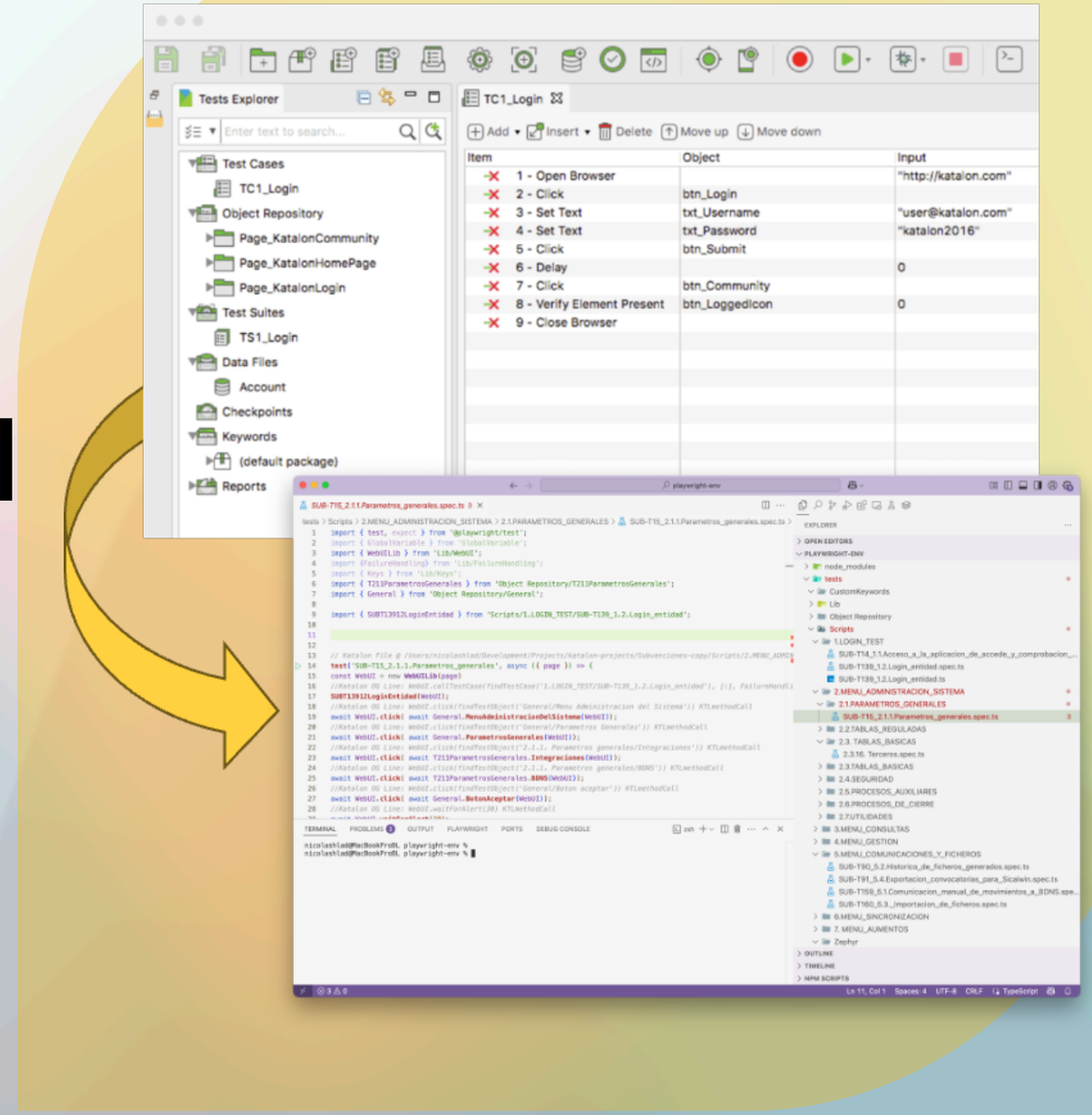
Migrating Katalon Studio Tests to Playwright with Model Driven Engineering



2nd of July 2025

Nicolas Hlad
Benoit Verhaeghe

What this presentation is about

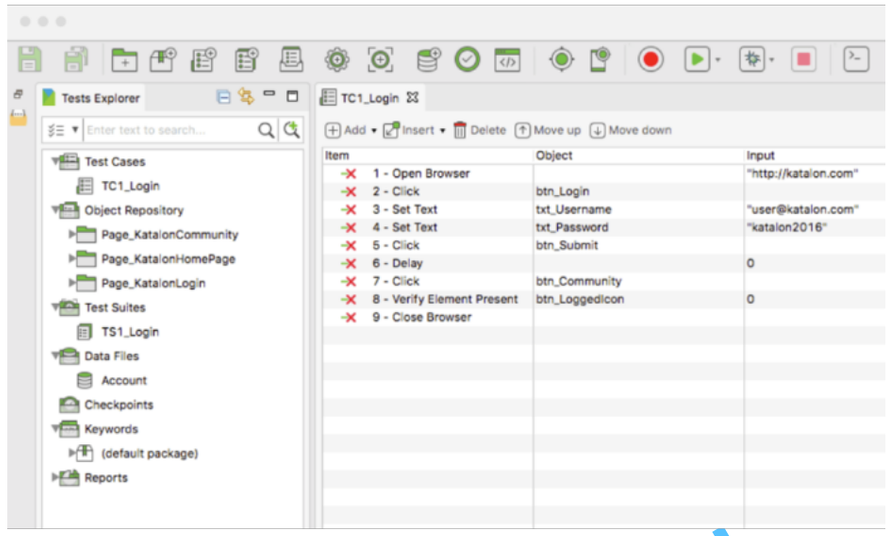
- An industrial use case of Pharo Moose
- Migration of test scripts using Model Driven Engineering (MDE).
- Our shared experience of
  - what worked
  - what didn't (kind of)

Introduction

1. Industrial context and problems
2. Defining a migration strategy
3. Designing our migration tool
4. Early results
5. Conclusion

Problem definition — 1

Migrating in time

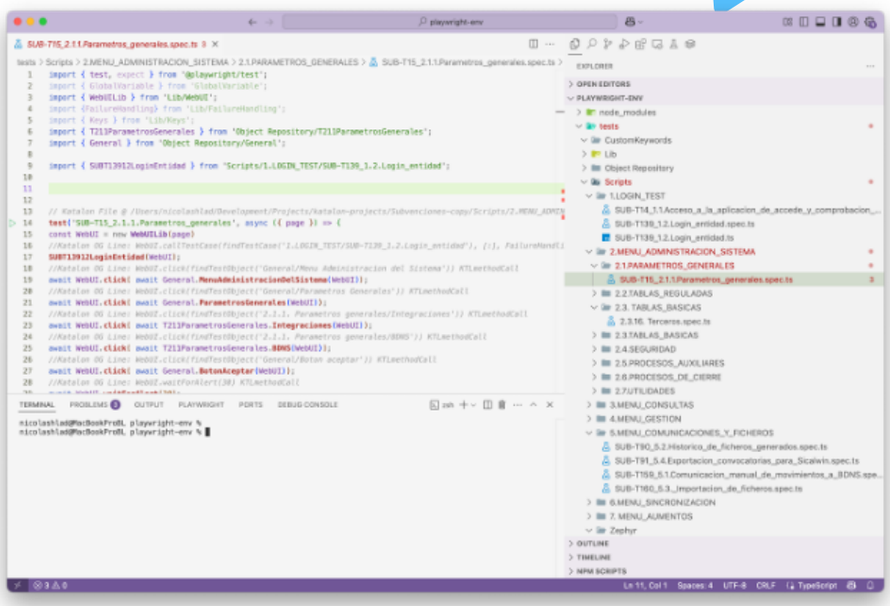


1. Four years ago, some Business Units (BUs) adopted *Katalon Studio* to implement their **functional behavior tests**

2. In **December 2024**, BUs were instructed to **change for *Playwright*** as a global solution for functional tests.

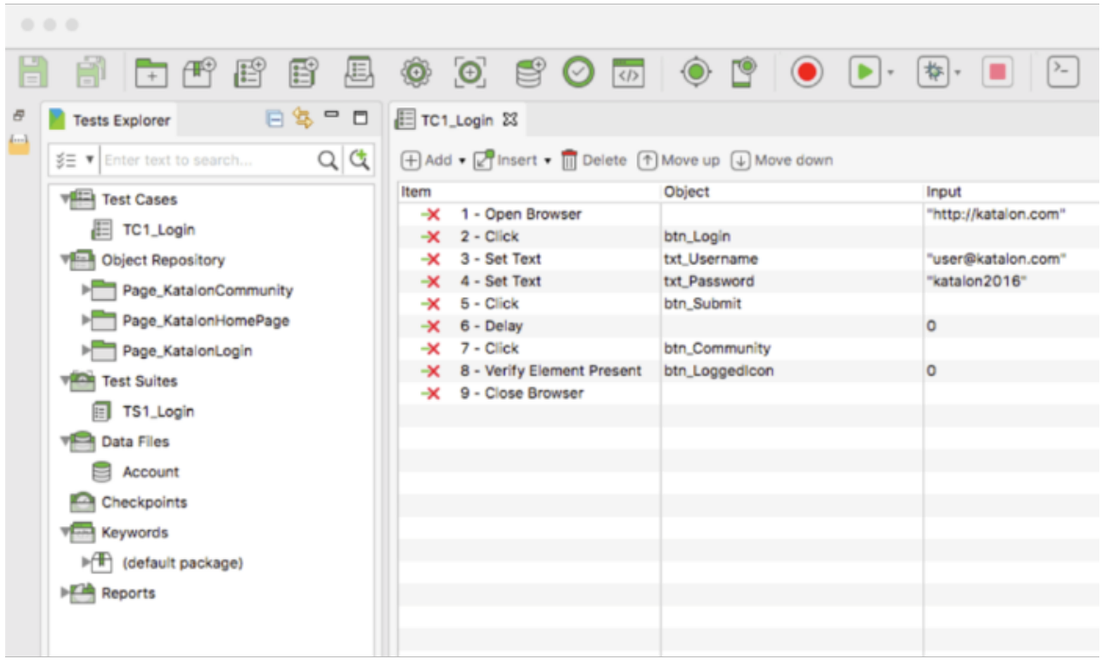
3. They had to migrate all tests **by June 2025**

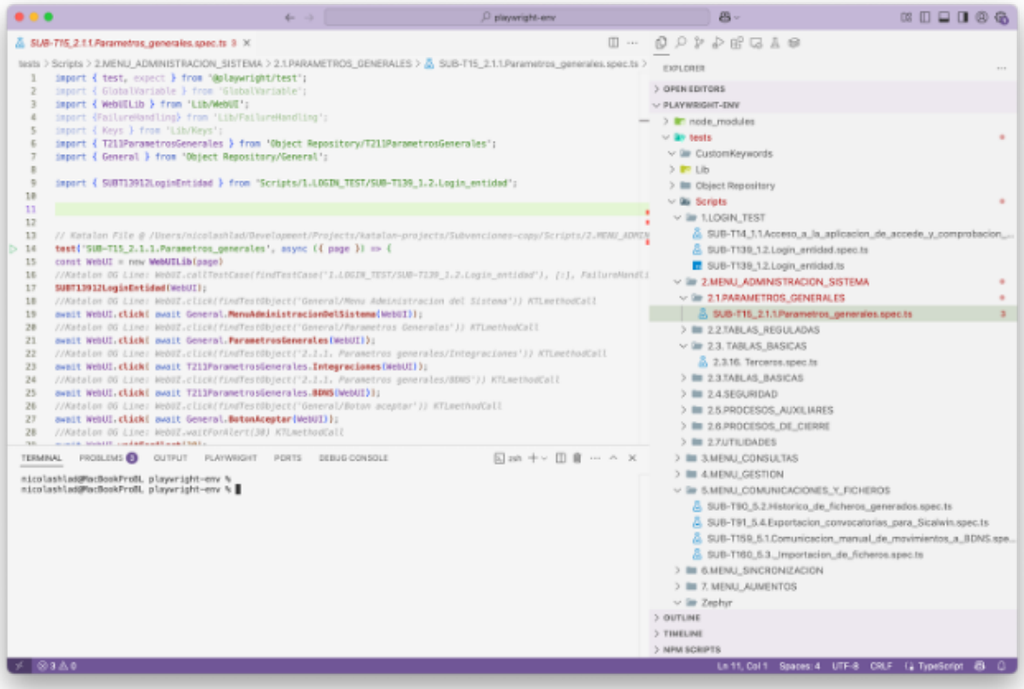
within 6 months



Katalon vs playwright

Katalon Studio is an EDI for testing web, mobile, API and desktop applications.





Playwright is a scripting framework for testing web, mobile, API and desktop applications.

Problem defintion — 3

From Katalon to Groovy

TC1_Login

Add ▾ Insert ▾ Delete Move up Move down

Item	Object	Input
1 - Open Browser		"http://katalon.com"
2 - Click	btn_Login	
3 - Set Text	txt_Username	"user@katalon.com"
4 - Set Text	txt_Password	"katalon2016"
5 - Click	btn_Submit	
6 - Delay		0
7 - Click	btn_Community	
8 - Verify Element Present	btn_LoggedIcon	0
9 - Close Browser		

```groovy
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI

// Step 1: Open Browser
WebUI.openBrowser('http://katalon.com')

// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))

// Step 3: Enter Username
WebUI.setText(findTestObject('txt_Username'), 'user@katalon.com')

// Step 4: Enter Password
WebUI.setText(findTestObject('txt_Password'), 'katalon2016')

// Step 5: Click Submit button
WebUI.click(findTestObject('btn_Submit'))

// Step 6: Delay for 0 seconds (can be skipped as it's 0)
WebUI.delay(0)

// Step 7: Click on Community button
WebUI.click(findTestObject('btn_Community'))

// Step 8: Verify Logged-in Icon is present
WebUI.verifyElementPresent(findTestObject('btn_LoggedIcon'), 0)

// Step 9: Close Browser
WebUI.closeBrowser()
```

*Fig —**Katalon Test script** can be exported into **source code** (Groovy)*

Problem defintion — 4

From Groovy… to Playwright ?

(a) ***Katalon***

```
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI

// Step 1: Open Browser
WebUI.openBrowser('http://katalon.com')

// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))

// Step 3: Enter Username
WebUI.setText(findTestObject('txt_Username'), 'user@katalon.com')

// Step 4: Enter Password
WebUI.setText(findTestObject('txt_Password'), 'katalon2016')

// Step 5: Click Submit button
WebUI.click(findTestObject('btn_Submit'))

// Step 6: Delay for 0 seconds (can be skipped as it's 0)
WebUI.delay(0)

// Step 7: Click on Community button
WebUI.click(findTestObject('btn_Community'))

// Step 8: Verify Logged-in Icon is present
WebUI.verifyElementPresent(findTestObject('btn_LoggedIcon'), 0)

// Step 9: Close Browser
WebUI.closeBrowser()
```

(b) ***Playwitght***

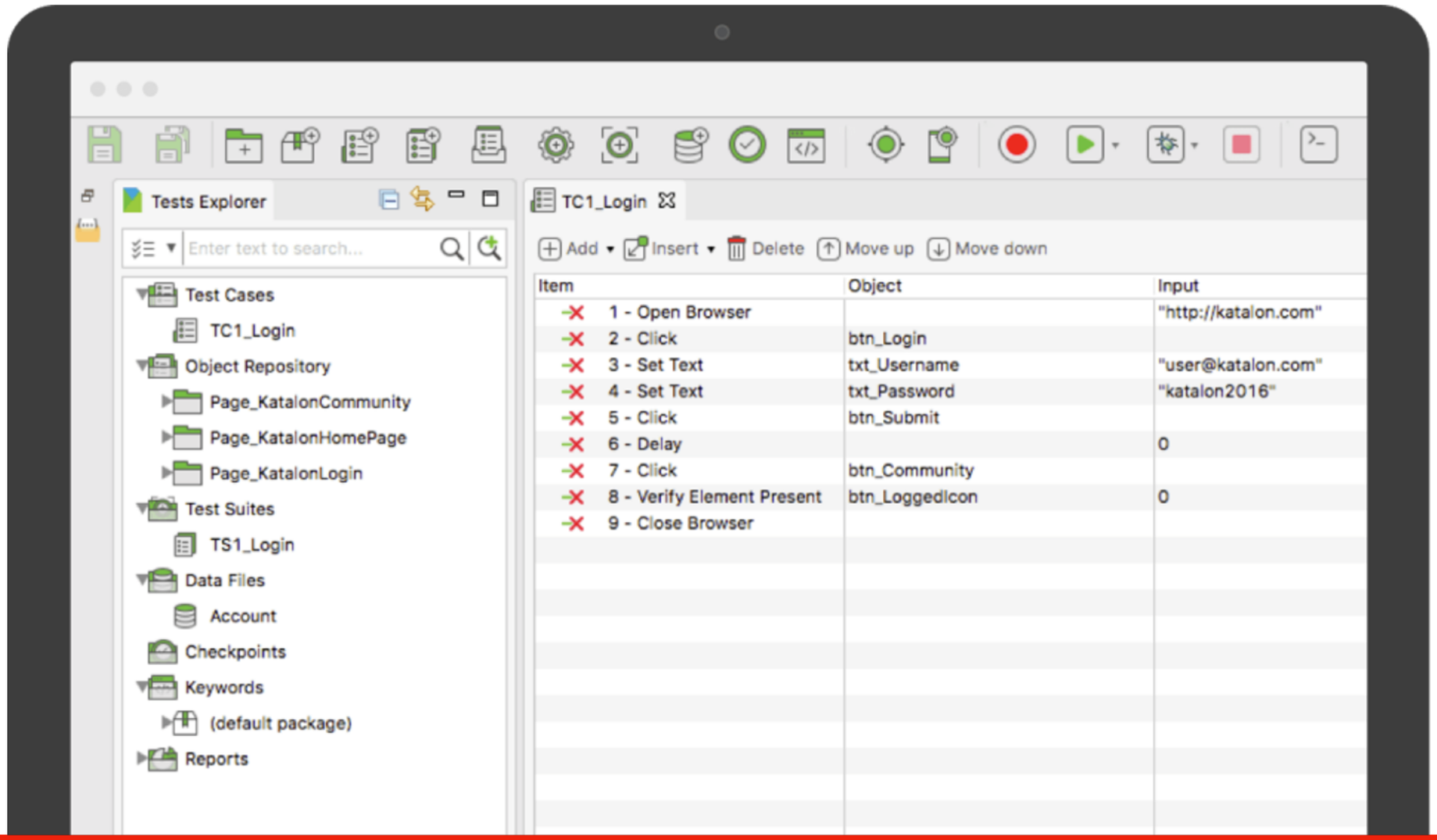
```
const { test, expect } = require('@playwright/test');

test('Login and verify icon in Playwright', async ({ page }) => {
  // Step 1: Open Browser and go to URL
  await page.goto('http://katalon.com');
  // Step 2: Click on Login button
  await page.click( await Login.BtnLogin(WebUI));
  // Step 3: Enter Username
  await page.fill('#txt_Username', 'user@katalon.com');
  // Step 4: Enter Password
  await page.fill('#txt_Password', 'katalon2016');
  // Step 5: Click Submit button
  await page.click(await Login.BtnSubmit(WebUI));
  // Step 6: Delay 0 seconds (optional, skipped)
  // Step 7: Click on Community button
  await page.click(await Login.BtnCommunity(WebUI));
  // Step 8: Verify Logged-in Icon is present
  await expect(page.locator('#btn_LoggedIcon')).toBeVisible();
  // Step 9: Close Browser — handled automatically by Playwright
});
```

Fig —A ***Katalon*** *that was re-written manually into a* ***Playwitght*** *test script (Typescript)*

Initialy, we were told that it would take "hours" to rewrite a test by hand

And we counted 1412 test cases

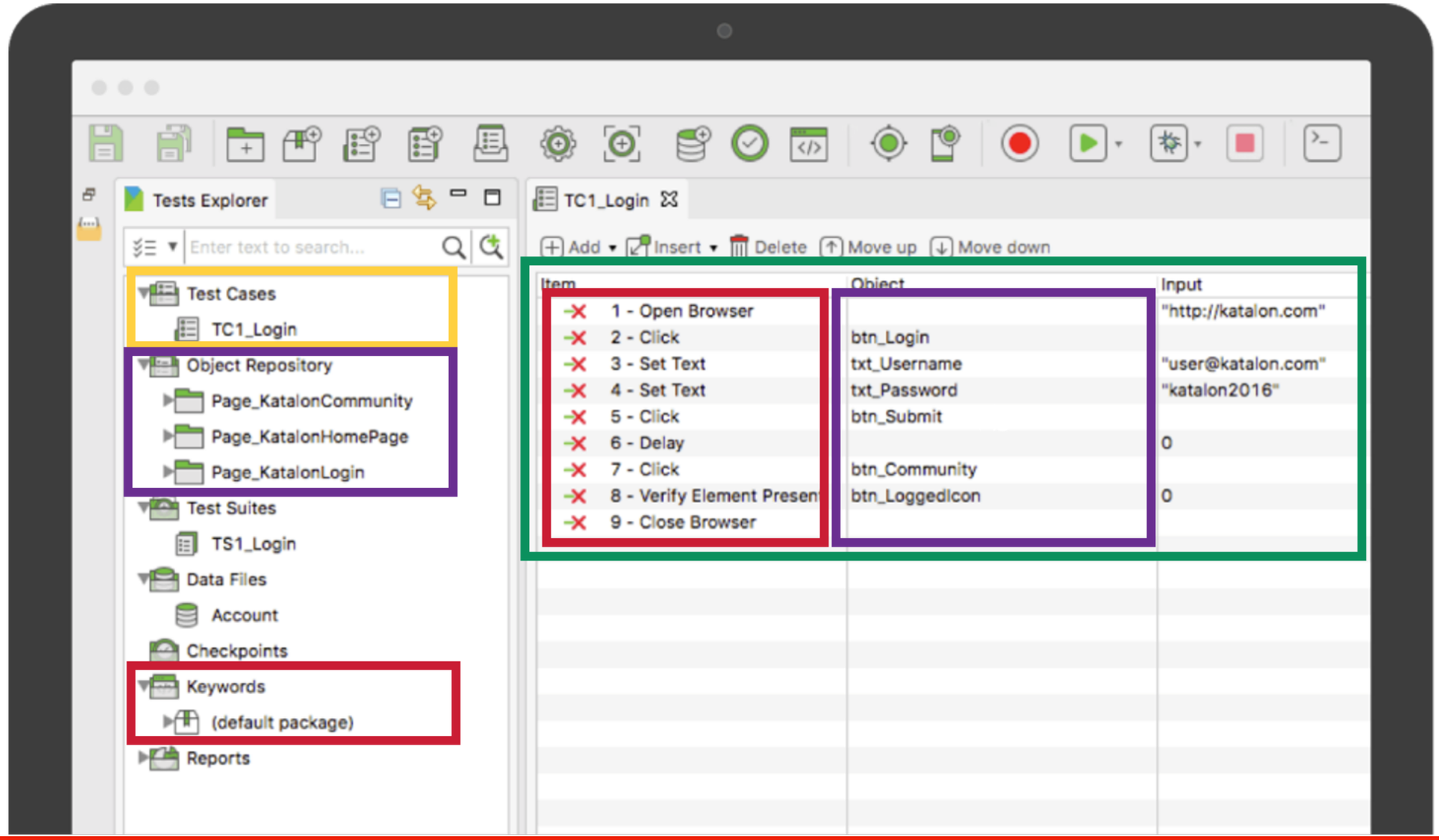
Defining concepts of Katalon's Tests



Katalon studio

Defining concepts of Catalan's Tests

Test Case	The *script* of a specific test
Test Step	An *instruction* in a Test case
Keyword	The *action* executed during a specific step (can have parameters)
Objects	The *receiver* of an action set by a keyword



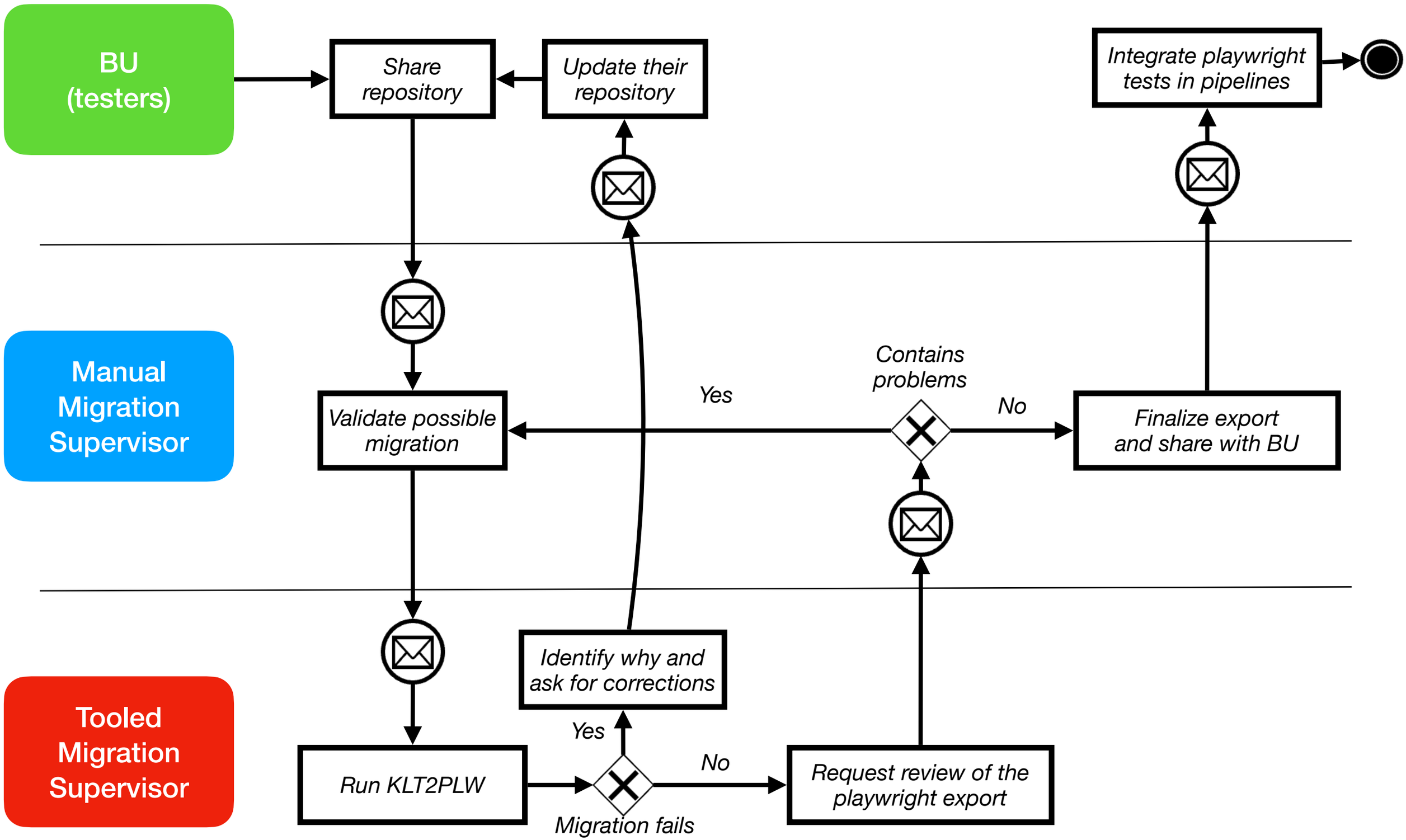
Additional challenges

- Testers used **Groovy** to declare their own **custom keyword**
- They wrote some tests using only Groovy, with **full Groovy syntax** (classes, loops, conditions, switch, etc).
- Some repositories were full of **emojis**, **accents**, and **errors** such as misaligned test cases between the Katalon Studio and the Groovy representation (save problems).

Planning the migration

- How to organise this migration between stakeholders ?
- How to migrate:
  - The language ? *from Groovy to Typescript*
  - The framework actions ? *Katalon's Keywords to Playwright's instructions*
- How to automate the migration ?

Collaborating on the migration



Migrating Language and Framework

(a) ***Katalon***

```
// Step 1: Open Browser
WebUI.openBrowser('http://katalon.com')

// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))
```

Groovy and Typescript are different languages but share some **similarities in their syntax**.

However, framework's actions are **mostly different**.

(b) ***typical*** *Playwright*

```
// Step 1: Open Browser and go to URL
await page.goto('http://katalon.com');
// Step 2: Click on Login button
await page.click( await Login.BtnLogin(WebUI));
```

Migrating Language and Framework

(a) ***Katalon***

```
// Step 1: Open Browser
WebUI.openBrowser('http://katalon.com')

// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))
```

Simple Keyword

Complexe Keyword

Groovy and Typescript are different languages but share some **similarities in their syntax**.

However, framework's actions are **completely different**.

(b) ***typical*** *Playwright*

```
// Step 1: Open Browser and go to URL
await page.goto('http://katalon.com');
// Step 2: Click on Login button
await page.click( await Login.BtnLogin(WebUI));
```

Simple keywords strategy

(a) ***Katalon***

```
// Step 1: Open Browser
WebUI.openBrowser('http://katalon.com')

// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))
```

(b) ***normal*** *Playwright*

```
// Step 1: Open Browser and go to URL
await page.goto('http://katalon.com');
```

(c) ***our*** *Playwright*

```
await WebUI.openBrowser('http://katalon.com');
```

WebUI.ts 30.50 KiB

```
export class  WebUILib {
    private page: Page;
    private context : BrowserContext;
    private browser : Browser | null;
```

We implemented 97 Keywords in the WebUI.ts

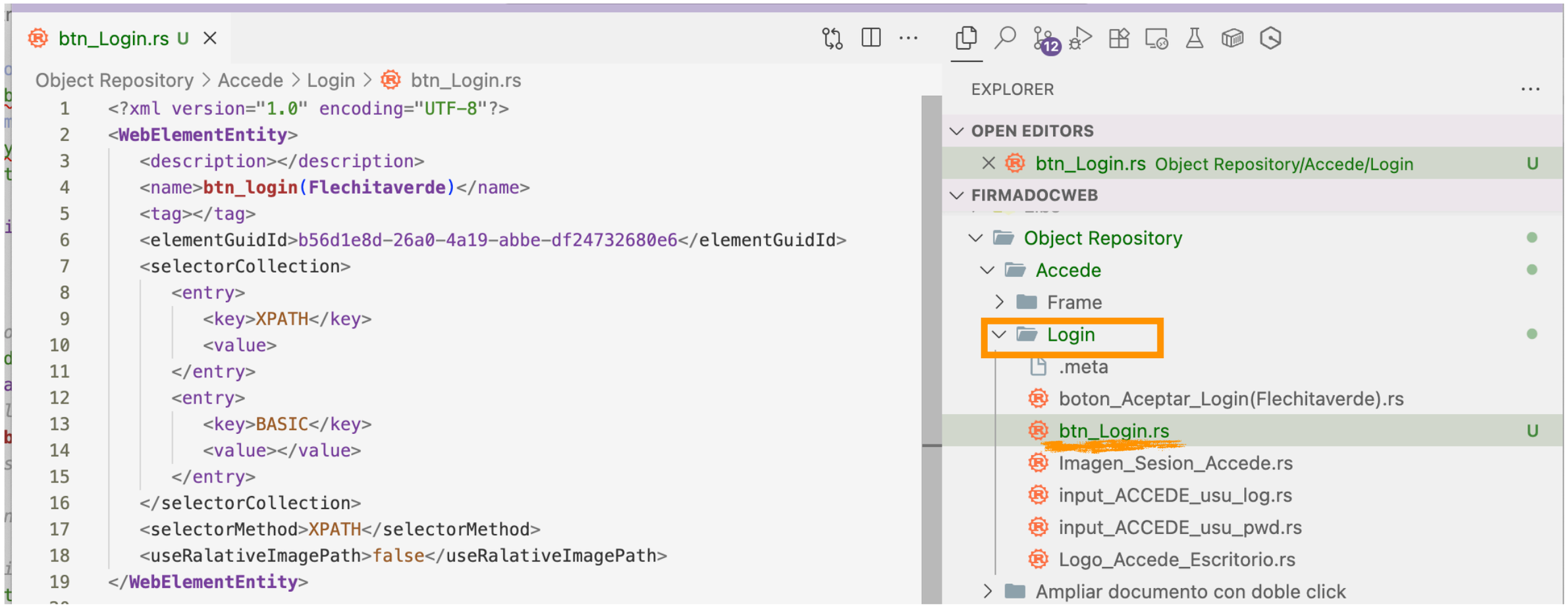
```
async openBrowser(url? : string,flowControl? : FailureHandling) {
    if (this.premiereOvertureFaite) {
        this.page = await this.context.newPage();
    }
    else {
        this.premiereOvertureFaite = true;
    }

    if(url != undefined && url.trim() != '') {
        await this.page.goto(url);
    }
}
```

Complexe keywords strategy

(a) ***Katalon***

```
// Step 1: Open Browser
WebUI.openBrowser('http://katalon.com')

// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))
```

1. Locating the objet definition file (in Katalon Repository)



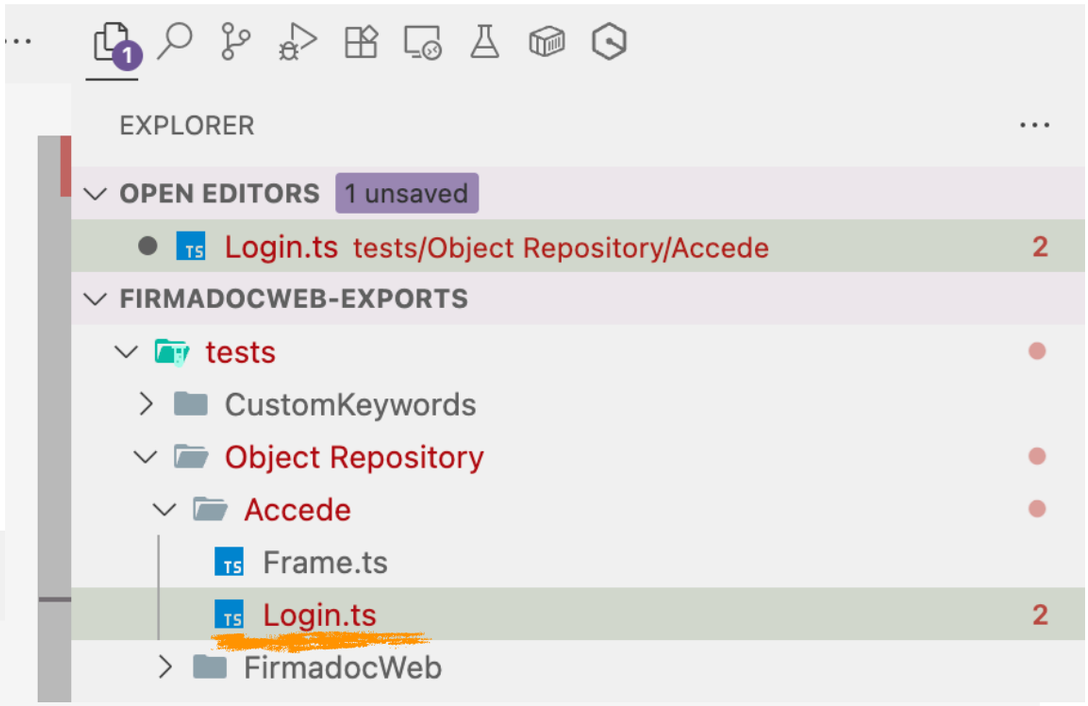
Migration

Complexe keywords strategy

(a) ***Katalon***

```
// Step 1: Open Browser
WebUI.openBrowser('http://katalon.com')

// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))
```

2. Create a Login TS object
    2.1. Matching folder name
    2.2. Exporting each Katalon Objects



Login.ts 2

tests > Object Repository > Accede > Login.ts > ...

```
1   import { Page } from '@playwright/test';
2   import { WebUILib } from 'Lib/WebUI';
3   import { Frame } from 'Object Repository/Accede/Frame';
4
5
6   export const Login = {
7   BtnLogin : async (WebUI : WebUILib) => {return WebUI.getPage().locator("//img[@id='idLogin']"); },
8   InputACCEDEUsuPwd : async (WebUI : WebUILib) => {return WebUI.getPage().locator("//input[@id='pwd']"); },
```

Complexe keywords strategy

(a) ***Katalon***

```
// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))
```

(b) ***our*** *Playwright*

```
// Step 2: Click on Login button
await WebUI.click(await Login.BtnLogin(WebUI));
```

Other cases like `findTestCase` our `custom keyword` are handle similarly

Katalon2Playwright (KTL2PLW)

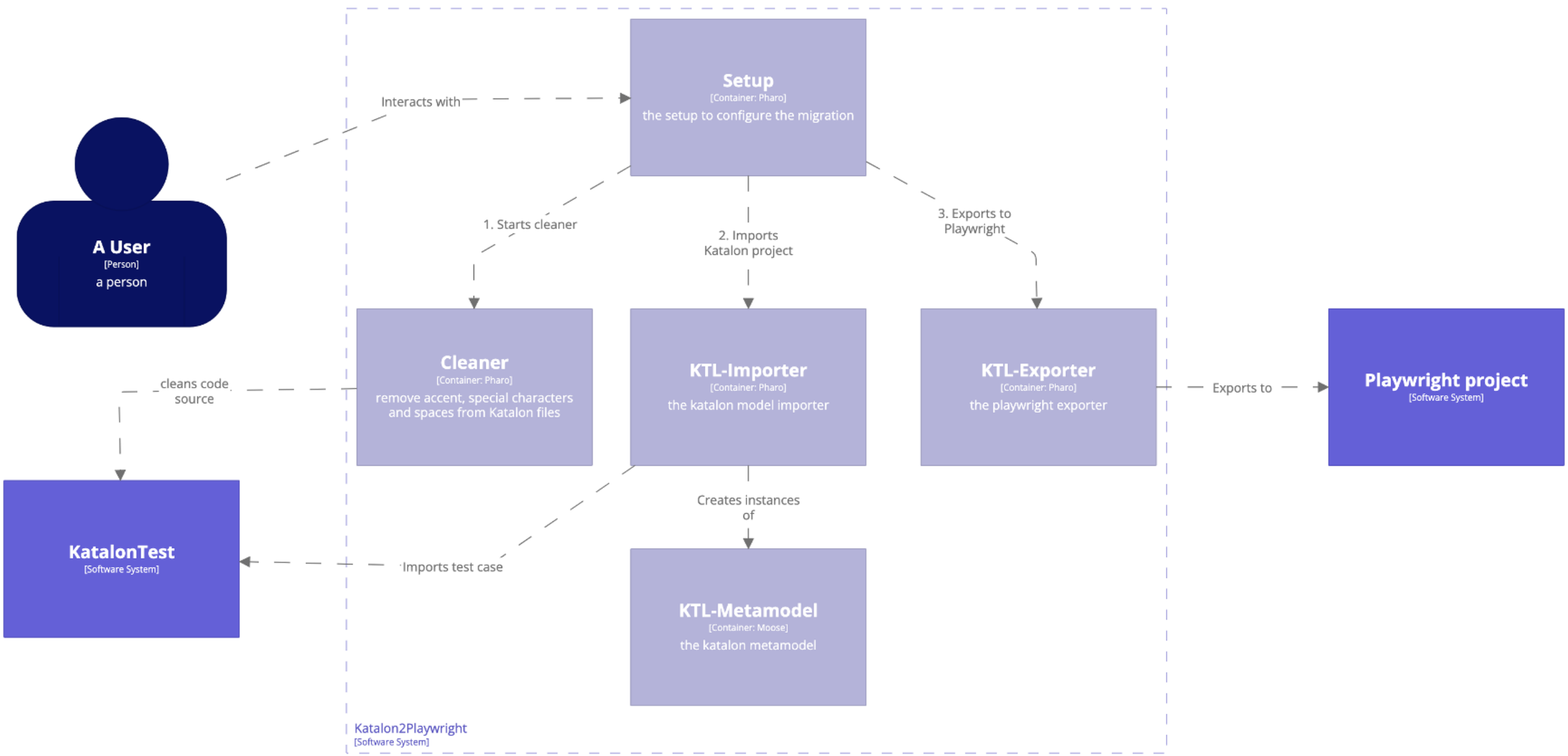


Fig — C4 Model of KTL2PLW usage

Cleaning repository

- Goal: to fix issues that cause *Error* when parsing the files
  - Remove accents from code and file names


```
@Keyword
public void  PestañaTramitacion(){
    WebUI.click(findTestObject('Object Repository/Firmadoc Web/Contratacion/Pestaña Tramitacion'))
}
```

- Transform Katalon “descriptions” into Groovy comments

```
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import internal.GlobalVariable as GlobalVariable

'Open browser and navigate to demo AUT site.'
WebUI.openBrowser(GlobalVariable.G_SiteURL)
```

Fig — A Katalon comment inside the groovy file representing a test case

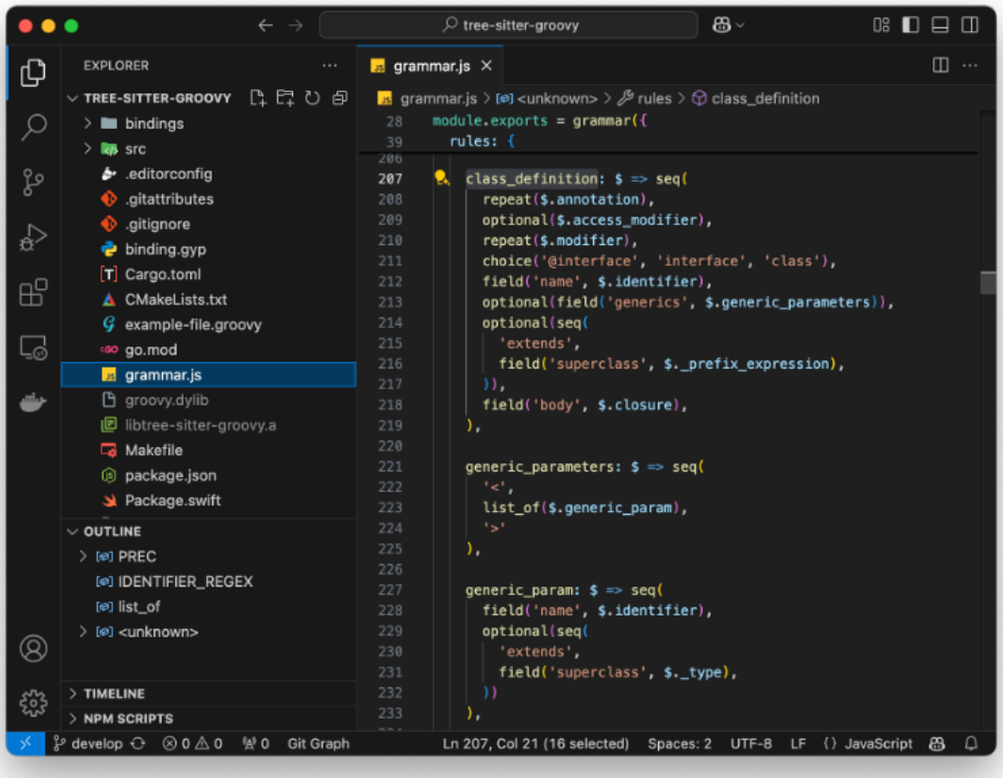
Migration tools in Pharo/Moose

Parsing with Tree-Sitter

Tree-Sitter is a C framework to build parser. It has a incremental construction of the AST.
It benefits from a large library of grammar from the community.

Providing an EBF grammar, it generates a parser as a lib of all major OS platforms.

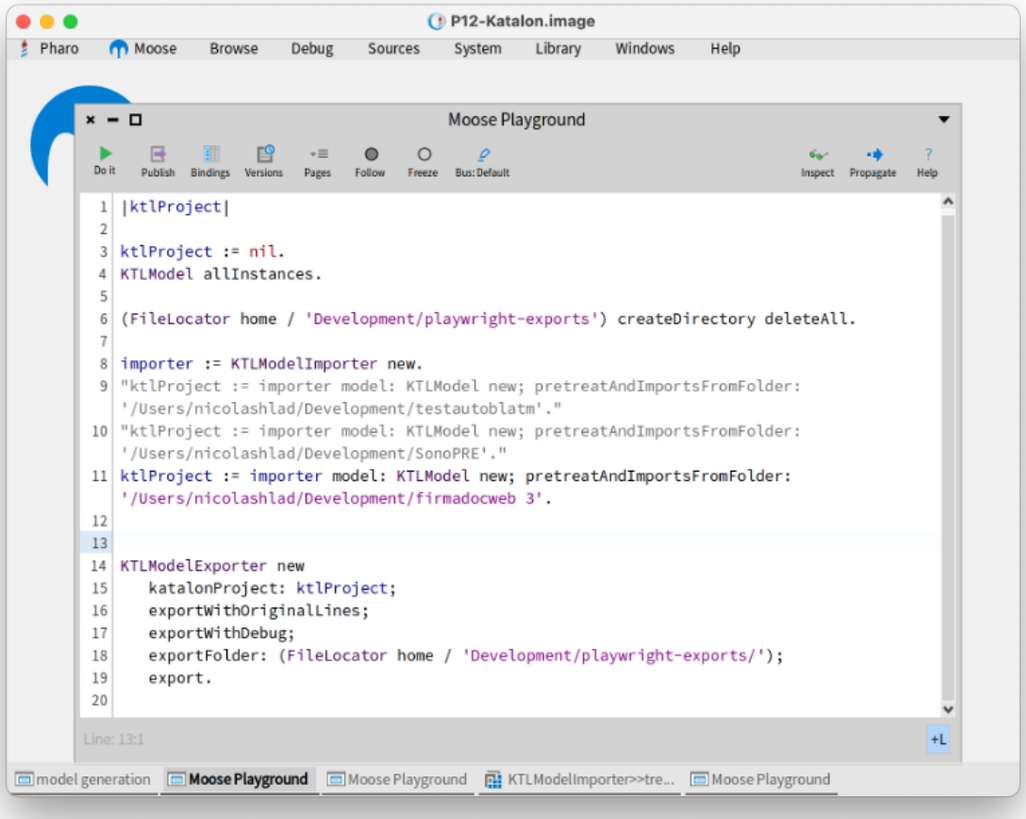
Tree-Sitter-Pharo (GitHub)



Tree-Sitter Groovy Parser (in C)

Foreign Function Interface (FFI)

communication



Pharo Image (smalltalk)

Modeling Katalon in Moose

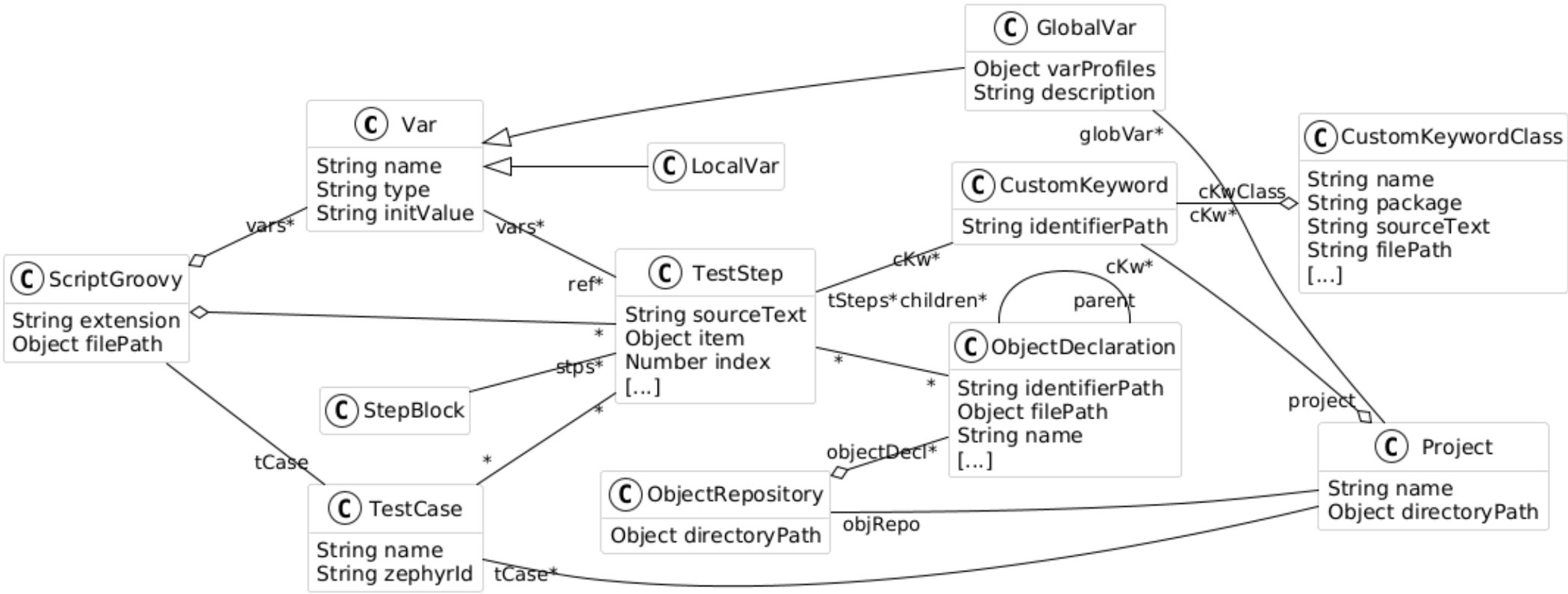


Fig — simplify Metamodel Katalon

Modeling Katalon in Moose

Test Case

Test Step

Keyword

Objects

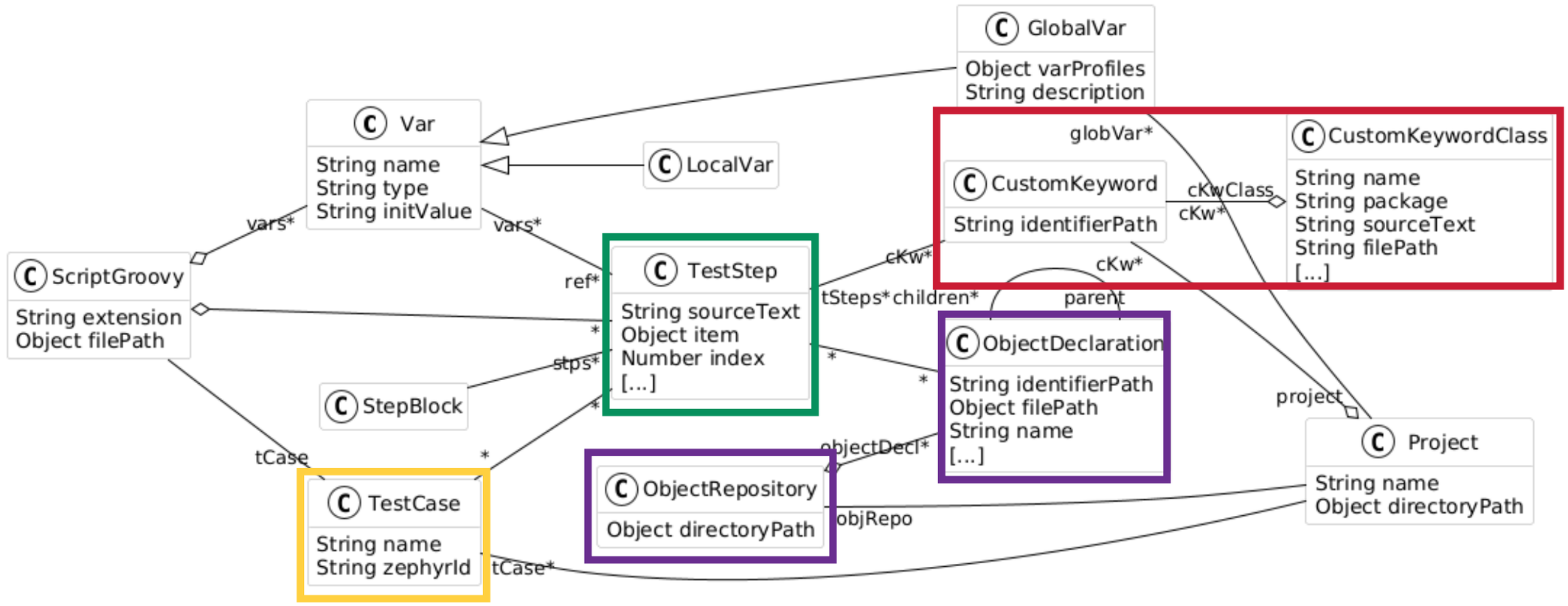
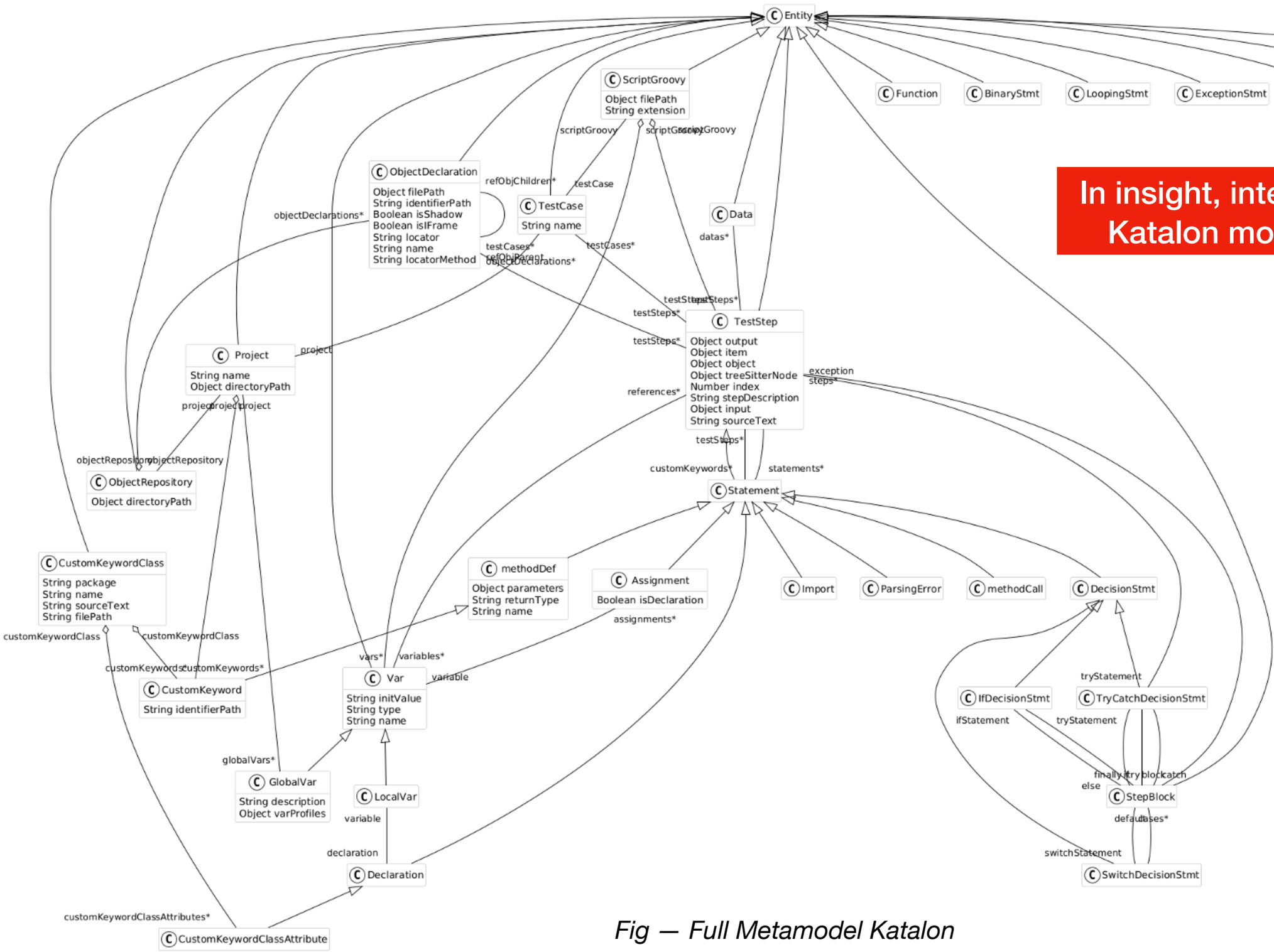


Fig — initial Metamodel Katalon

However, it lacks source declaration of *custom keyword* or the use of Groovy *statement* in Test Steps

KTLModel



In insight, integrating Groovy into Katalon model was a mistake

Fig — Full Metamodel Katalon

Running on a medium size project

Items	Raw	Breakpoints	Meta

Key	Value
objectRepositoryFiles	829
testCaseFiles	160
customKeywordsFiles	10

Fig — Firmadoc project description

a Bag [7613 items]

Items	Raw	Breakpoints	Meta

Items	Occurences
KTLStatement	84
KTLLocalVar	79
KTLGlobalVar	17
KTLParsingError	1
KTLDeclaration	52
KTLDecisionStmt	1
KTLObjectRepository	1
KTLmethodCall	3081
KTLProject	1
KTLImport	2930
KTLScriptGroovy	160
KTLAssignment	29
KTLCustomKeywordClassAttribute	2
KTLTestCase	160
KTLIfDecisionStmt	20
KTLObjectDeclaration	828
KTLTryCatchDecisionStmt	1
KTLSwitchDecisionStmt	1
KTLCustomKeyword	118
KTLStepBlock	37
KTLCustomKeywordClass	10

Fig — KTL model's instance for Firmadoc

Note:

- 829 object files vs. 828 object declarations (i.e. one empty file)
- 10 Custom Keywords files for 118 Custom Keyword (i.e. multiple keywords per file)
- 513 MB project migrated in ~10sec (i.e. not all files interested)

Early results

project	Object	TC[2]	Ckw[3]	Migr[4]	#Pb_BMC[5]	#Pb_AMC[6]
paReg	378	44	6	113276	1385	605
paPart	1239	116	0	9332	84	82
paUrb	410	39	0	4217	101	41
paAdm	136	13	0	5880	71	68
paEmp	5	1	0	4179	0	0
paRen	1649	118	0	24852	1056	365
sec	433	69	0	6763	219	24
csaKvP	785	408	0	13908	98	32
taBl	642	171	0	6897	297	240
subv	486	98	0	9630	496	496
fdw	829	160	10	10565	427	297
pruBas	512	109	1	3295	563	313
notiFica	441	66	0	6126	101	5
Average				16840.0	375.2	197.5

Table 1
Results of the migrated Katalon projects at Berger-Levrault. [2]Test Case; [3]Custom Keyword; [4]Migration Time in msec; [5]Problems Before Manual Checking; [6]remaining Problems After Manual Checking

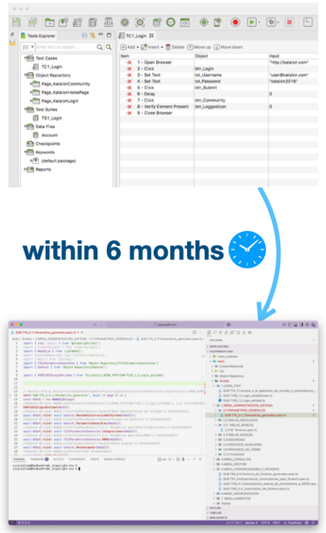
We computed an estimated **8512 hours** of manual migration for all test cases only. With Ktl2Plw, we migrated all projects in **218920 msecs.**

Conclusion

Industrial context

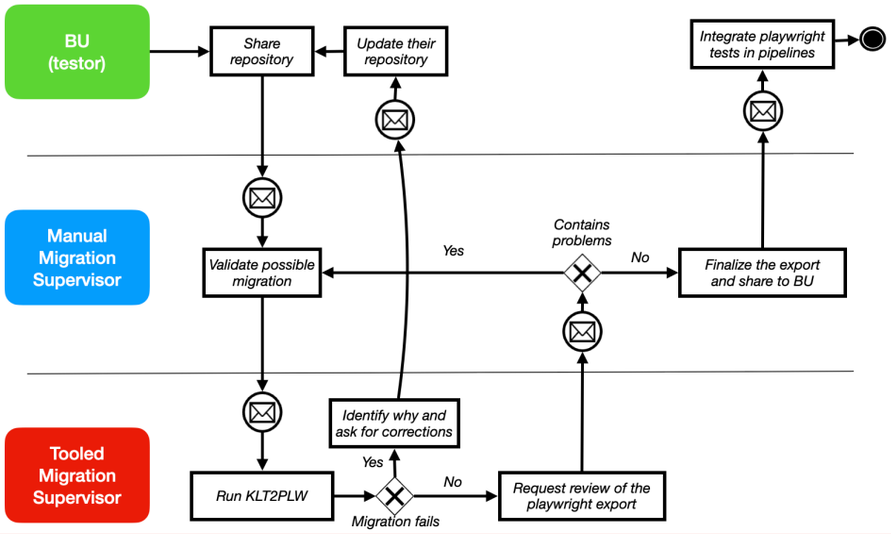
Problem definition — 1

1. Four years ago, some Business Unit (Bu) adopted to Katalon studio to implement their **functional behavior tests**
2. In December 2024, BU were notified to adopt playwright as a global solution for functional tests.
3. Thus, BU had to migrate to Playwright. The migration need to happen **before June 2025**



Migration

Collaborating on the migration



Tooled Migration

Modeling Katalon in Moose

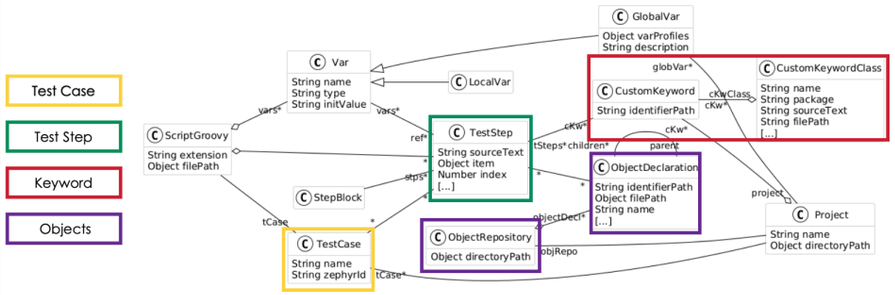


Fig — initial Metamodel Katalon

However, it lacks source declaration of *custom keyword* or the use of Groovy *statement* in Test Steps

Results

Early results

project	Object	TC[2]	Ckw[3]	Migr[4]	#Pb_BMC[5]	#Pb_AMC[6]
paReg	378	44	6	113276	1385	605
paPart	1239	116	0	9332	84	82
paUrb	410	39	0	4217	101	41
paAdm	136	13	0	5880	71	68
paEmp	5	1	0	4179	0	0
paRen	1649	118	0	24852	1056	365
sec	433	69	0	6763	219	24
csaKvP	785	408	0	13908	98	32
taBl	642	171	0	6897	297	240
subv	486	98	0	9630	496	496
fdw	829	160	10	10565	427	297
pruBas	512	109	1	3295	563	313
notiFica	441	66	0	6126	101	5
Average				16840.0	375.2	197.5

Table 1
Results of the migrated Katalon projects at Berger-Levrault. [2]Test Case; [3]Custom Keyword; [4]Migration Time in msec; [5]Problems Before Manual Checking; [6]remaining Problems After Manual Checking

We computed an estimated **85123 hours** of manual migration for all test cases only. With Ktl2Plw, we migrated all projects in **218920 msecs.**

Lessons learned

Take away messages

- Migrating code involves communication
  - Involve **a lot** of feedback loops

- For MDE, divide the concepts into sub-models
  - separate Model Framework from Model language (Katalon / Groovy)
  - Model can be connected together later on (moose connector)

- Migrate step by step
  - Deliver and iterate to add more features

Annexe

Script1725613040251.groovy

```groovy
import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint
import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase
import static com.kms.katalon.core.testdata.TestDataFactory.findTestData
import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject
import static com.kms.katalon.core.testobject.ObjectRepository.findWindowsObject
import com.kms.katalon.core.checkpoint.Checkpoint as Checkpoint
import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as CucumberKW
import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile
import com.kms.katalon.core.model.FailureHandling as FailureHandling
import com.kms.katalon.core.testcase.TestCase as TestCase
import com.kms.katalon.core.testdata.TestData as TestData
import com.kms.katalon.core.testng.keyword.TestNGBuiltinKeywords as TestNGKW
import com.kms.katalon.core.testobject.TestObject as TestObject
import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows
import internal.GlobalVariable as GlobalVariable
import org.openqa.selenium.Keys as Keys

WebUI.callTestCase(findTestCase('FirmadocWeb/Sesion/Cambio_contraseña_en_Firmadoc_Web_DEPRECATED'), [:], FailureHandling.STOP_ON_FAILURE)
WebUI.closeBrowser()
WebUI.openBrowser('')
WebUI.navigateToUrl('http://url:8080/login.do')
WebUI.maximizeWindow()
WebUI.navigateToUrl('http://url:8080/jsp/endpoint')
WebUI.setText(findTestObject('Object Repository/Accede/Login/input_ACCEDE_usu_log'), 'ADM')
WebUI.setEncryptedText(findTestObject('Accede/Login/input_ACCEDE_usu_login'), 'xxxxxxx')
WebUI.sendKeys(findTestObject('Object Repository/Accede/Login/input_ACCEDE_usu_login'), Keys.chord(Keys.ENTER))
```

IYA.17._Entrar_en_Accede_con_cambio_contrasena_en_Firmadoc_Web_DEPRECATED.spec.ts

```typescript
import { test, expect } from '@playwright/test';
import { GlobalVariable } from 'GlobalVariable';
import { WebUILib } from 'Lib/WebUI';
import {FailureHandling} from 'Lib/FailureHandling';
import { Keys } from 'Lib/Keys';
import { Login } from 'Object Repository/Accede/Login';

import { CambioContrasenaEnFirmadocWebDEPRECATED } from 'Scripts/FirmadocWeb/Sesion/Cambio_contrasena_en_Firmadoc_Web_DEPRECATED';

test('IYA.17._Entrar_en_Accede_con_cambio_contrasena_en_Firmadoc_Web_DEPRECATED', async ({ page }) => {
    const WebUI = new WebUILib(page)
    CambioContrasenaEnFirmadocWebDEPRECATED(WebUI);
    await WebUI.closeBrowser();
    await WebUI.openBrowser('');
    await WebUI.navigateToUrl('http://url:8080/login.do');
    await WebUI.maximizeWindow();
    await WebUI.navigateToUrl('http://url:8080/jsp/endpoint');
    await WebUI.setText( await Login.InputACCEDEUsuLog(WebUI), 'ADM');
    await WebUI.setEncryptedText( await Login.InputACCEDEUsuLogin(WebUI), 'xxxxxxx');
    await WebUI.sendKeys( await Login.InputACCEDEUsuLogin(WebUI), Keys.chord(Keys.ENTER));
});
```

Texte du titre

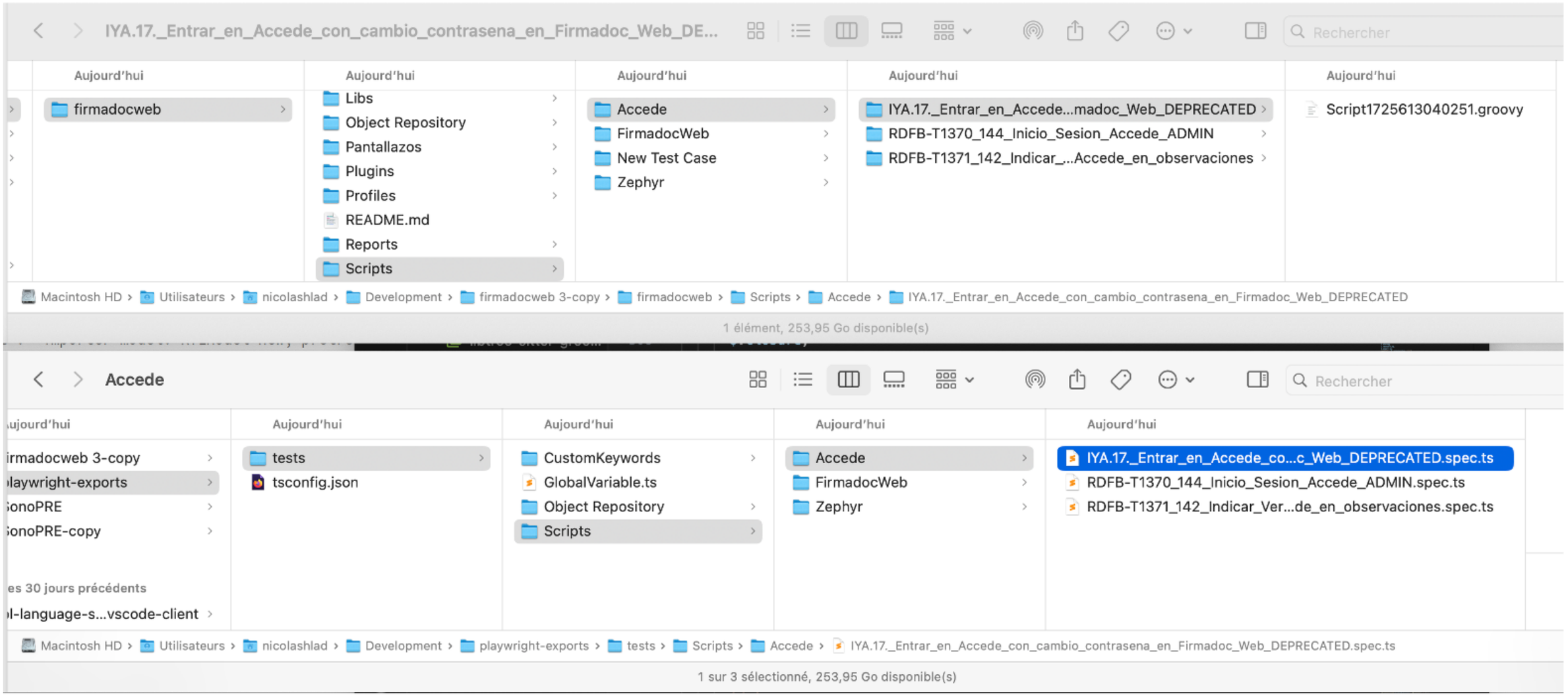


Fig — Retaining file structure while migrating scripts

Users > nicolashlad > Development > firmadocweb 3 > firmadocweb > Scripts > Accede > IYA.17._Entrar_en_Accede_con_cambio_contraseña_en_Firmadoc_Web_DEPRECATED > Script1725613040251.groovy

```groovy
import com.kms.katalon.core.checkpoint.Checkpoint as Checkpoint
import com.kms.katalon.core.cucumber.keyword.CucumberBuiltinKeywords as CucumberKW
import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile
import com.kms.katalon.core.model.FailureHandling as FailureHandling
import com.kms.katalon.core.testcase.TestCase as TestCase
import com.kms.katalon.core.testdata.TestData as TestData
import com.kms.katalon.core.testng.keyword.TestNGBuiltinKeywords as TestNGKW
import com.kms.katalon.core.testobject.TestObject as TestObject
import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords as WS
import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords as WebUI
import com.kms.katalon.core.windows.keyword.WindowsBuiltinKeywords as Windows
import internal.GlobalVariable as GlobalVariable
import org.openqa.selenium.Keys as Keys

WebUI.callTestCase(findTestCase('FirmadocWeb/Sesion/Cambio_contraseña_en_Firmadoc_Web_DEPRECATED'), [:], FailureHandling.STOP_ON_FAILURE)
WebUI.closeBrowser()
WebUI.openBrowser('')
WebUI.navigateToUrl('http://blcersel:8080/login.do')
WebUI.maximizeWindow()
WebUI.navigateToUrl('http://blcersel:8080/jsp/administracion/login.jsp?modo=4&nocache=0.012509764395485456')
WebUI.setText(findTestObject('Object Repository/Accede/Login/input_ACCEDE_usu_log'), 'ADM')
WebUI.setEncryptedText(findTestObject('Accede/Login/input_ACCEDE_usu_pwd'), 'V0DQ9szQkFM=')
WebUI.sendKeys(findTestObject('Object Repository/Accede/Login/input_ACCEDE_usu_pwd'), Keys.chord(Keys.ENTER))
```

From Katalon Test Case …

… to a Playwright Test

tests > Scripts > Accede > IYA.17._Entrar_en_Accede_con_cambio_contrasena_en_Firmadoc_Web_DEPRECATED.spec.ts > ...

```typescript
import { test, expect } from '@playwright/test';
import { GlobalVariable } from 'GlobalVariable';
import { WebUILib } from 'Lib/WebUI';
import {FailureHandling} from 'Lib/FailureHandling';
import { Keys } from 'Lib/Keys';
import { Login } from 'Object Repository/Accede/Login';
import { CambioContrasenaEnFirmadocWebDEPRECATED } from 'Scripts/FirmadocWeb/Sesion/Cambio_contrasena_en_Firmadoc_Web_DEPRECATED.spec';

test('IYA.17._Entrar_en_Accede_con_cambio_contrasena_en_Firmadoc_Web_DEPRECATED', async ({ page }) => {
const WebUI = new WebUILib(page)
//Katalon OG Line: WebUI.callTestCase(findTestCase('FirmadocWeb/Sesion/Cambio_contrasena_en_Firmadoc_Web_DEPRECATED'), [:], FailureHandling.STOP_ON_FAILURE) KTLmethodCall
 CambioContrasenaEnFirmadocWebDEPRECATED(WebUI);
//Katalon OG Line: WebUI.closeBrowser() KTLmethodCall
 await WebUI.closeBrowser();
//Katalon OG Line: WebUI.openBrowser('') KTLmethodCall
 await WebUI.openBrowser('');
//Katalon OG Line: WebUI.navigateToUrl('http://blcersel:8080/login.do') KTLmethodCall
 await WebUI.navigateToUrl('http://blcersel:8080/login.do');
//Katalon OG Line: WebUI.maximizeWindow() KTLmethodCall
 await WebUI.maximizeWindow();
//Katalon OG Line: WebUI.navigateToUrl('http://blcersel:8080/jsp/administracion/login.jsp?modo=4&nocache=0.012509764395485456') KTLmethodCall
 await WebUI.navigateToUrl('http://blcersel:8080/jsp/administracion/login.jsp?modo=4&nocache=0.012509764395485456');
//Katalon OG Line: WebUI.setText(findTestObject('Object Repository/Accede/Login/input_ACCEDE_usu_log'), 'ADM') KTLmethodCall
  await WebUI.setText( await Login.InputACCEDEUsuLog(WebUI), 'ADM');
//Katalon OG Line: WebUI.setEncryptedText(findTestObject('Accede/Login/input_ACCEDE_usu_pwd'), 'V0DQ9szQkFM=') KTLmethodCall
  await WebUI.setEncryptedText( await Login.InputACCEDEUsuPwd(WebUI), 'V0DQ9szQkFM=');
//Katalon OG Line: WebUI.sendKeys(findTestObject('Object Repository/Accede/Login/input_ACCEDE_usu_pwd'), Keys.chord(Keys.ENTER)) KTLmethodCall
  await WebUI.sendKeys( await Login.InputACCEDEUsuPwd(WebUI), Keys.chord(Keys.ENTER));
});
```

Migrating complexe keyword

(a) ***Katalon***

```
// Step 1: Open Browser
WebUI.openBrowser('http://katalon.com')

// Step 2: Click on Login button
WebUI.click(findTestObject('btn_Login'))
```

WebUI.ts 30.50 KiB

```
async openBrowser(url? : string,flowControl? : FailureHandling) {
    if (this.premiereOvertureFaite) {
        this.page = await this.context.newPage();
    }
    else {
        this.premiereOvertureFaite = true;
    }

    if(url != undefined && url.trim() != '') {
        await this.page.goto(url);
    }
}
```

```
async click(locator : Locator ,flowControl? : FailureHandling) {
    await locator.click();
}
```

Tooled migration

KTLModel