



*Agility. Security. Delivered.*

# Leveraging Zephyr and Behave for Test Case Management

Jonathan Miller Kauffman

July 18, 2017

# Project Background

---

- **Client Type:** Bio-medical device company.
- **Engagement Type:** Agile assessment and transformation.
- **Engagement Length:** ~1 year
- **Product Being Developed:** Console that monitors/controls a heart pump that provides circulatory support to a patient.
- **Constraints:** Console was an embedded system. Client was working in an FDA-regulated environment.
- **My Role:** Support the transformation of the testing team: setting up frameworks, working within the team, and providing informal coaching.

# Problem: Initial State of Testing/Automation



## The Bad

- Monolithic Word documents.
- Late development caused rushed testing.
- Testing almost entirely manual.

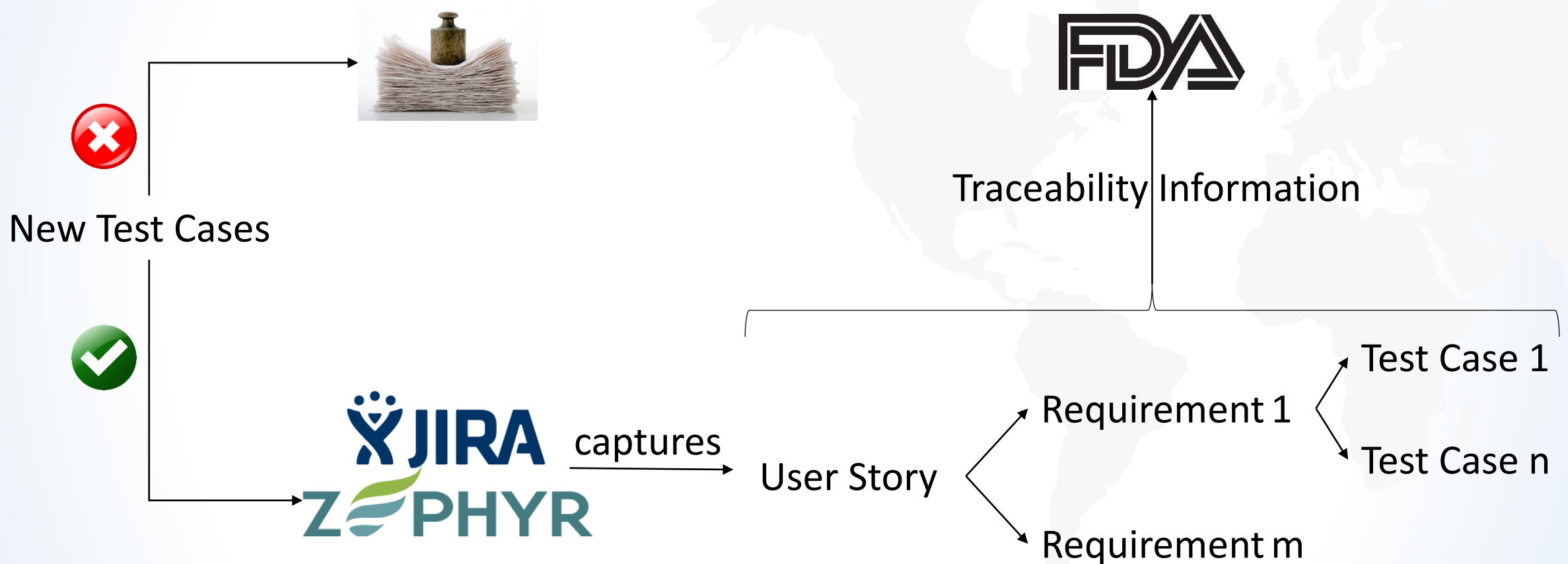


## The Promising

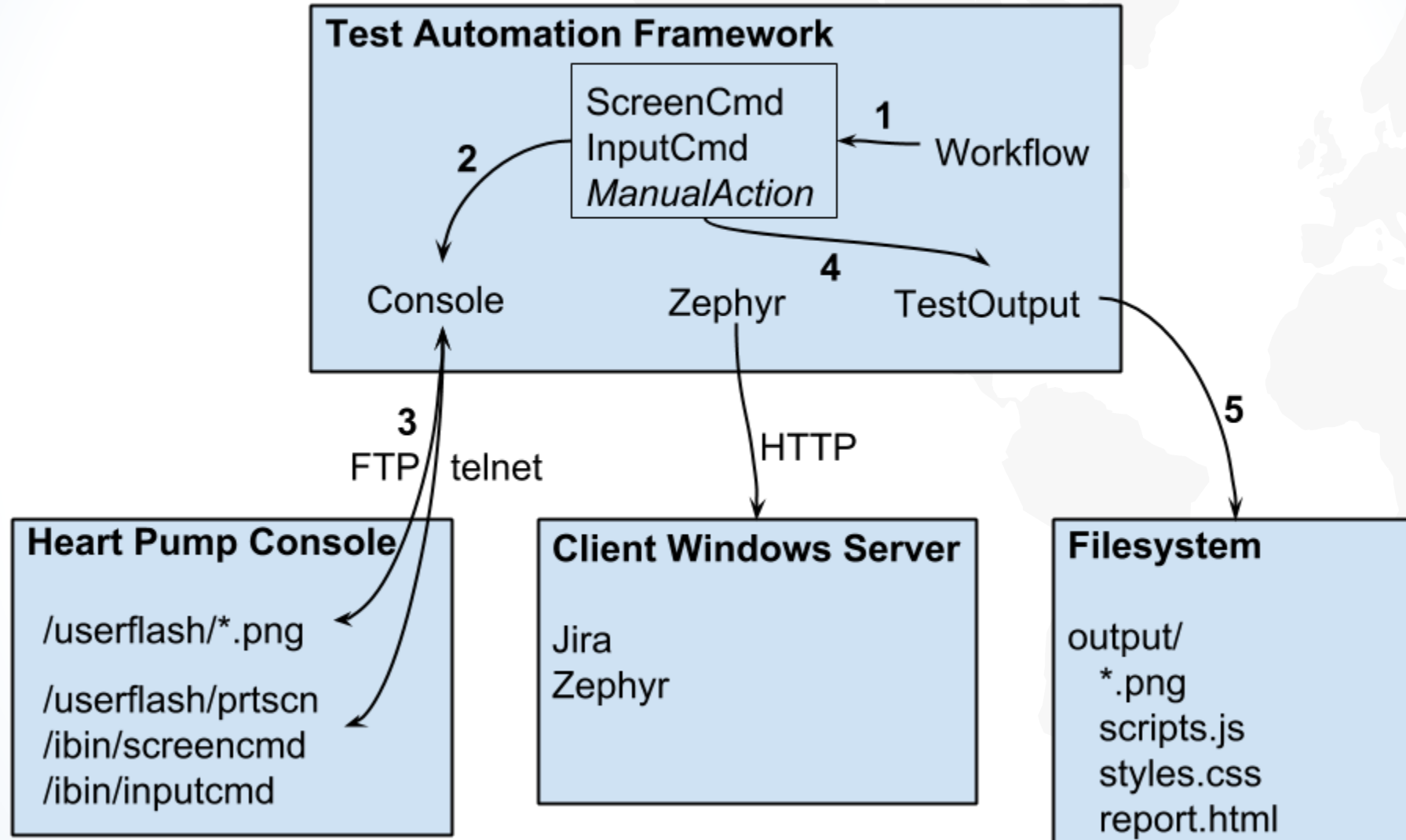
- Tools existed to facilitate automated testing.
- Limited automation existed (e.g., testing for memory leaks).



# First: Start Using JIRA/Zephyr



# Next: Create Test Automation Framework



# Next: Make That Framework “Behave”

## Feature File (features/example.feature)

Scenario

Given *pre-condition*.

When *action*.

Then *post-condition*.

## Step File (steps/example.py)

```
@given_behavior('pre-condition')  
def step_impl(context):  
    context.screencmd.go_to_home_screen()
```

```
@when_behavior('action')  
def step_impl(context):  
    context.inputcmd.press_rotary_knob()
```

```
@then_behavior('post-condition')  
def step_impl(context):  
    context.test_output.recordThenStep()
```

# Next: Write Test Results to Zephyr

Test Automation Framework



JIRA630 / JIRA-113  
NAME\_OF\_TESTCASE

Edit Comment Start Progress Done Admin Clone More Actions Execute

Details

Type: Test Status: **TO DO** (View Workflow)  
Priority: Major Resolution: Unresolved  
Affects Version/s: None Fix Version/s: None  
Component/s: None  
Labels: None

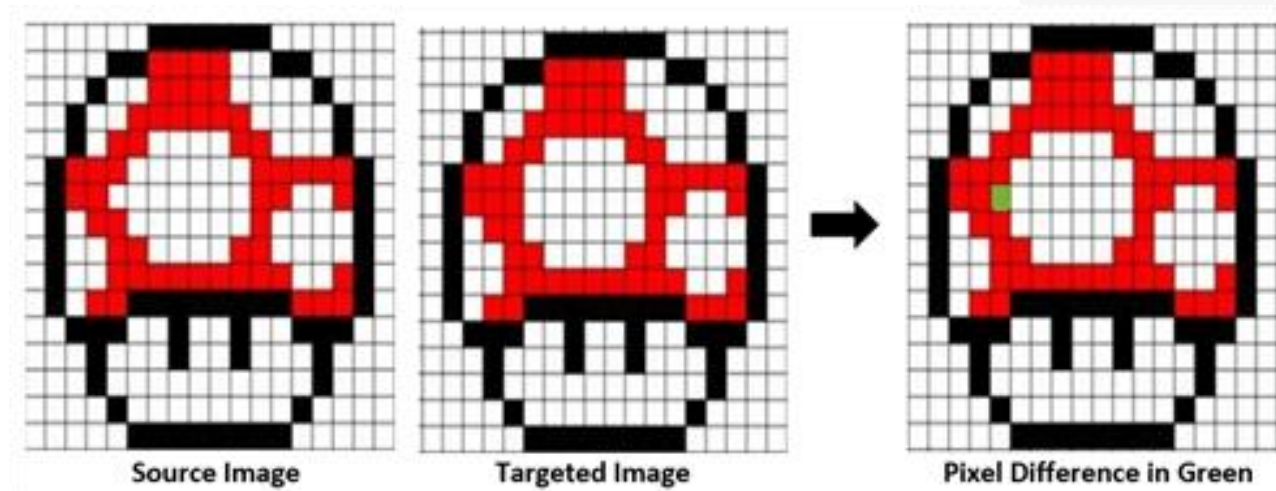
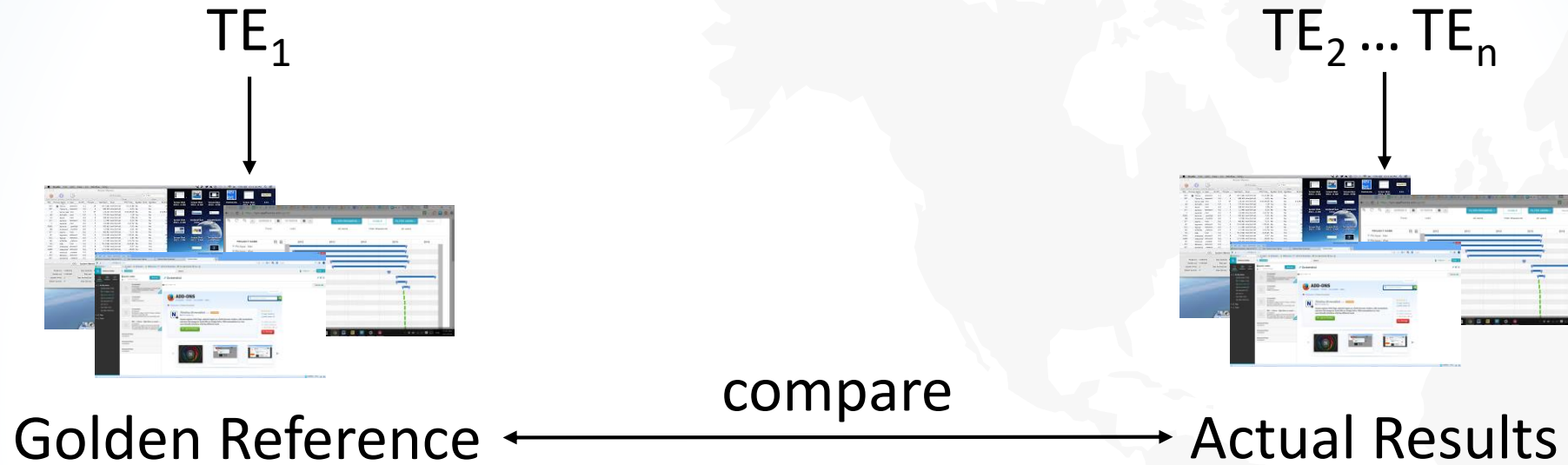
> Description

Test Details

Test Step	Test Data	Expected Result
1 Do a thing		This is what to expect
2 Do another thing		more expected results
		<input type="text"/>

Add

# Next: Automated Test Result Verification



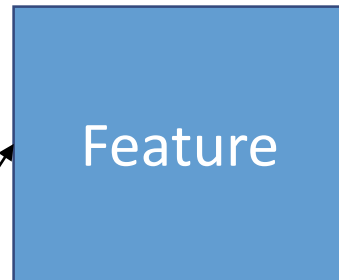


# Next: Zephyr Test Steps as Source of Truth



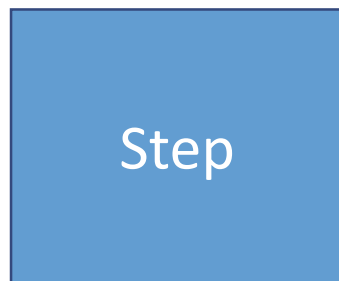
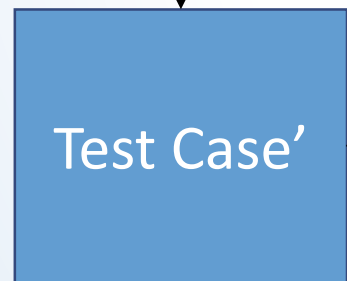
Before

ZEPHYR



change

change

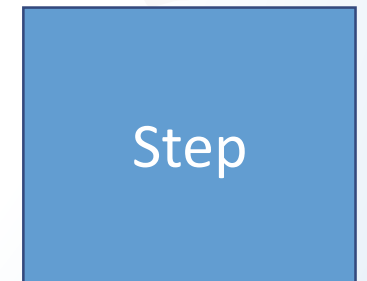
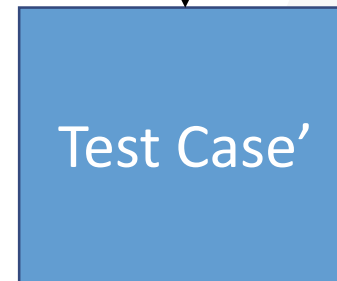


After

ZEPHYR



change



# Generating Feature Files for a Test Case

## Test Development



PROJECT-123

Feature File

Test Step 1

Given a console.

Step Description → When **SD-1.**

Expected Result → Then **ER-1.**

Test Data → And **TD-1.**

Test Step 2

When **SD-2.**

Step Description → Then **ER-2.**

Expected Result →

## DevOps Pipeline



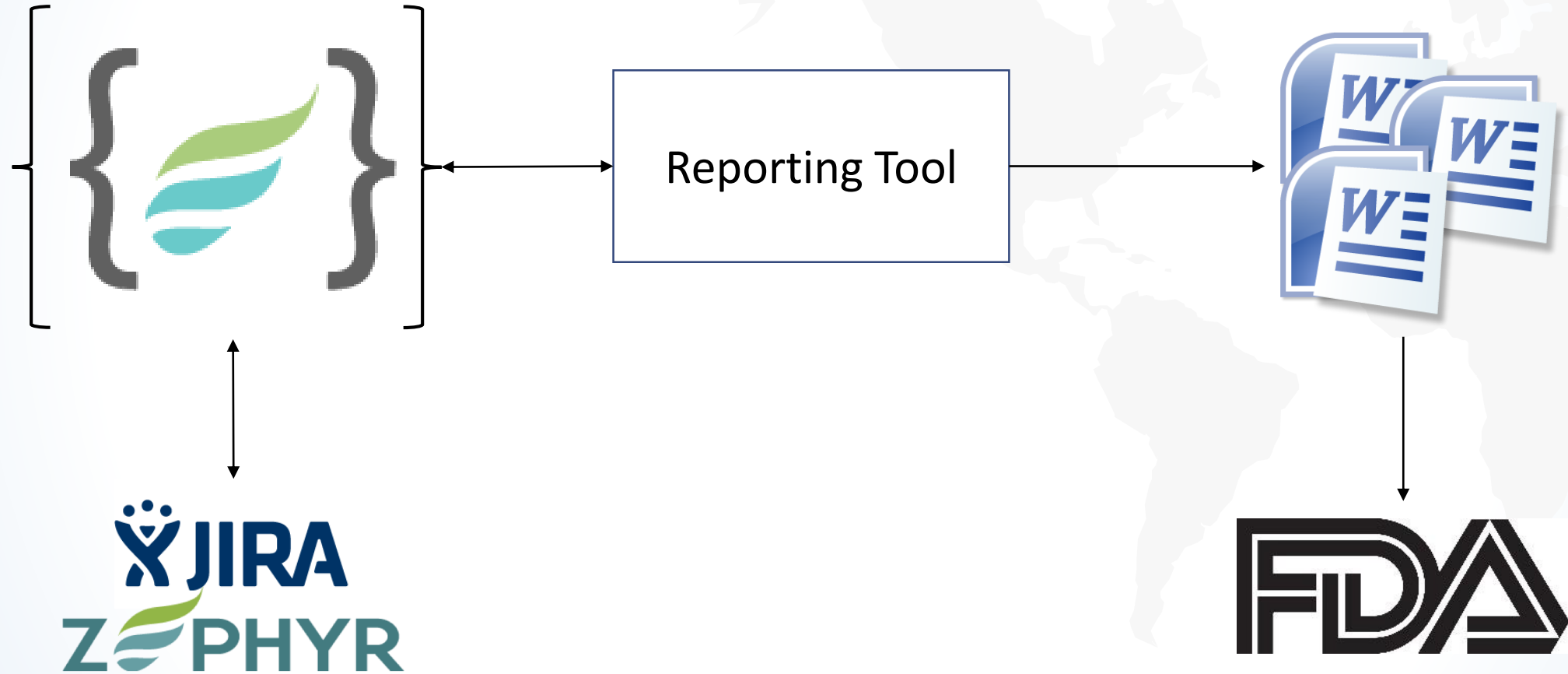
Automation ready

Test Case 1 → Feature 1

Test Case 2 → Feature 2

Test Case 3 → Feature 3

# Reporting: FDA Submission Documentation



# The End: Final State of Testing



## Jenkins

User Story

Requirement 1  
Requirement m

Test Case 1  
Test Case n

Step Implementations

1. Generate feature files.
2. Pull latest step implementations.
3. Run tests.

Reporting Tool

per release



# Takeaways (#Coveros5)

---

- Automation may threaten manual testers' sense of job security.
- Implement changes to testing practices in digestible chunks in order to allow team members to adapt to those changes.
- Comparing text strings is generally preferable to image verification, as it tends to be fragile.
- Value can still be gained by automating some subset of tests without having a hardware simulation capability.
- Maintain the steps for a test case in one place if both manual and automated test management tools are being used.

A faint, light-colored world map is visible in the background of the slide, centered behind the text.

# Questions?