CME ART

Automated Regression Testing

An ART for software testing

2016

Version 0.6

Confidential

Revision History

Date	Version	Description	A*, M, D	Prepared by
Dec 05, 2015	0.1	Initial version	A*	Phi Hoang
Aug 05, 2016	0.2	Add more info	A*, M	Phi Hoang
Aug 11, 2016	0.3	Add description about modules in the ART.	A*, M	Phi Hoang
Aug 14, 2016	0.4	Add general functionalities, business rules and update quick start.	A*, M	Phi Hoang
Aug 28, 2016	0.5	Update how keyword applied	A*, M	Phi Hoang
Sep 9, 2016	0.6	Add package release structure	A*, M	Phi Hoang

Contents

1. Introduction

This document is a summary of technical design and philosophies applied in an automation test framework created by CME.TESHCM. The framework is named ART which has been developed since Dec, 2014.

Let's review some reasons WHY we actually need to have automation testing, especially the ART.

Improves accuracy and quality

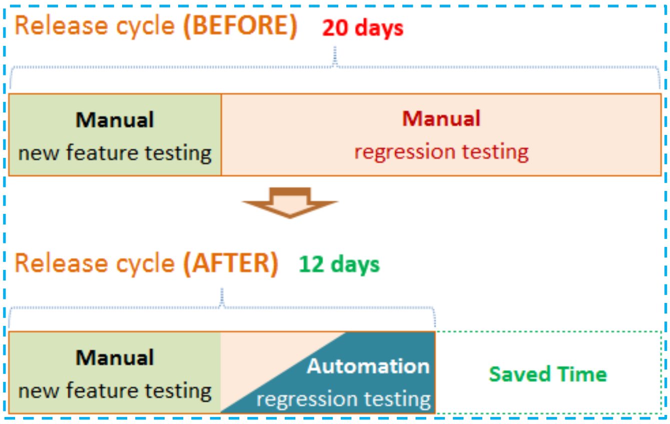
- **Reliability** – ensure correct test at protocol layer and **eliminate human error**.
- **Auditability** – provide comprehensive results.
- **Regression** testing – gain confidence that nothing is broken.

Increases test coverage

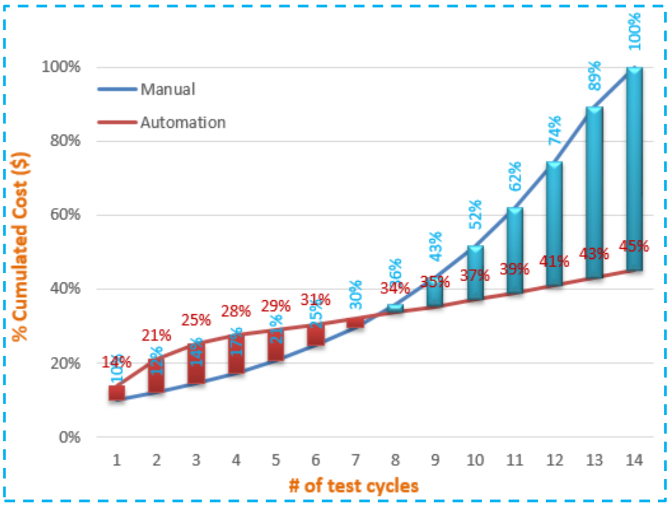
- **Huge data verification achievable only by automation.** (i.e. verify thousands of XML attributes)
- **Better coverage**, more executions, higher number of test cycles.

Saves time and money

- **Reduce execution time** up to **40%**.



- **Save Cost** – compared to only manual testing.



2. Terminologies and Abbreviations

Term	Description
ART	Automated Regression Testing, indicate the automation test framework created by CME.TESHCM.
CME	Communication, Media and Entertainment.

CI/CD	Continuous Integration / Continuous Deployment (Delivery)
AUT	Application Under Test

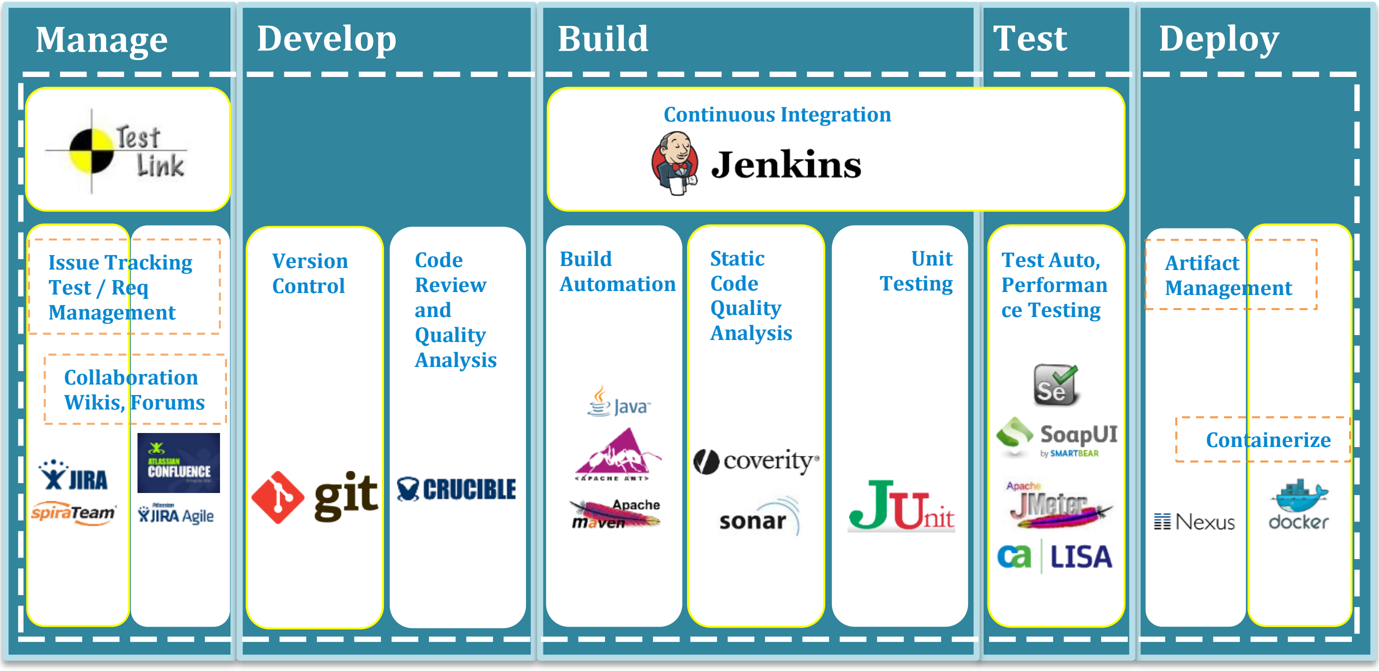
Abbreviation	Description
$	The root folder of the ART project suite.

3. Testing Problems and High Level Solutions

#	Testing problem	High Level Solution
1	• Must align testing with agile development methodologies which require short time to market.	• Do automation testing and integrate it to **Continuous Integration**.
2	• Defragmented testing result so it is difficult to make decision.	• Standardize proper technology stack to **centralize testing result** from both manual and automation teams.
3	• High data rates and responsiveness. • Different technology platforms under test (OS, DB, Web Server and application protocols and development).	• Design automation test framework to deal with **Shell, Web Service, Web Application, gigabytes of XML Data** on **Windows** and **Linux.**

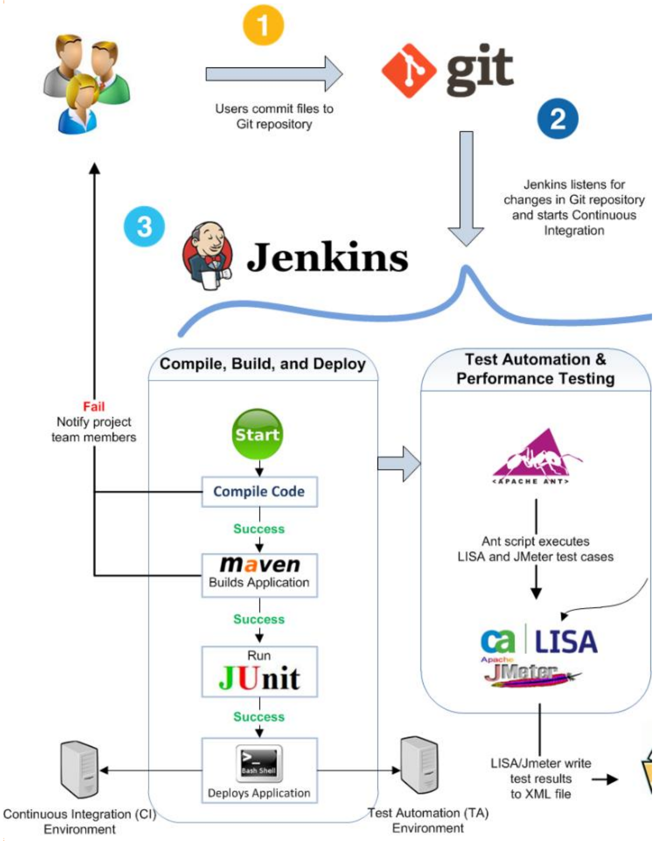
4. Standardized Tools in CI/CD

Automation testing is only a part of quality activities and it is recommended to align with development processes and tools.

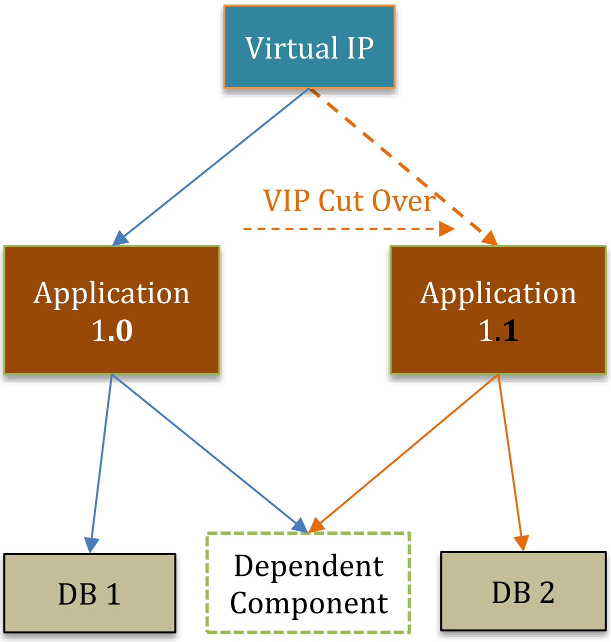


4.1. Automate Deployment (Continuous Deployment)

- By **saving time** in deployment, testers will be able to spend more time on feature testing.
- **Prevent mistakes** in installation while there is lots of configuration defined in release note.



- New and old application/services are deployed side by side to **eliminate delay impact** on testing team in case testing is blocked on the new version.
- Testing team **starts testing cycle** on the new version after **smoke testing passed**.



4.2. Automate Test integrated with Continuous Integration tool

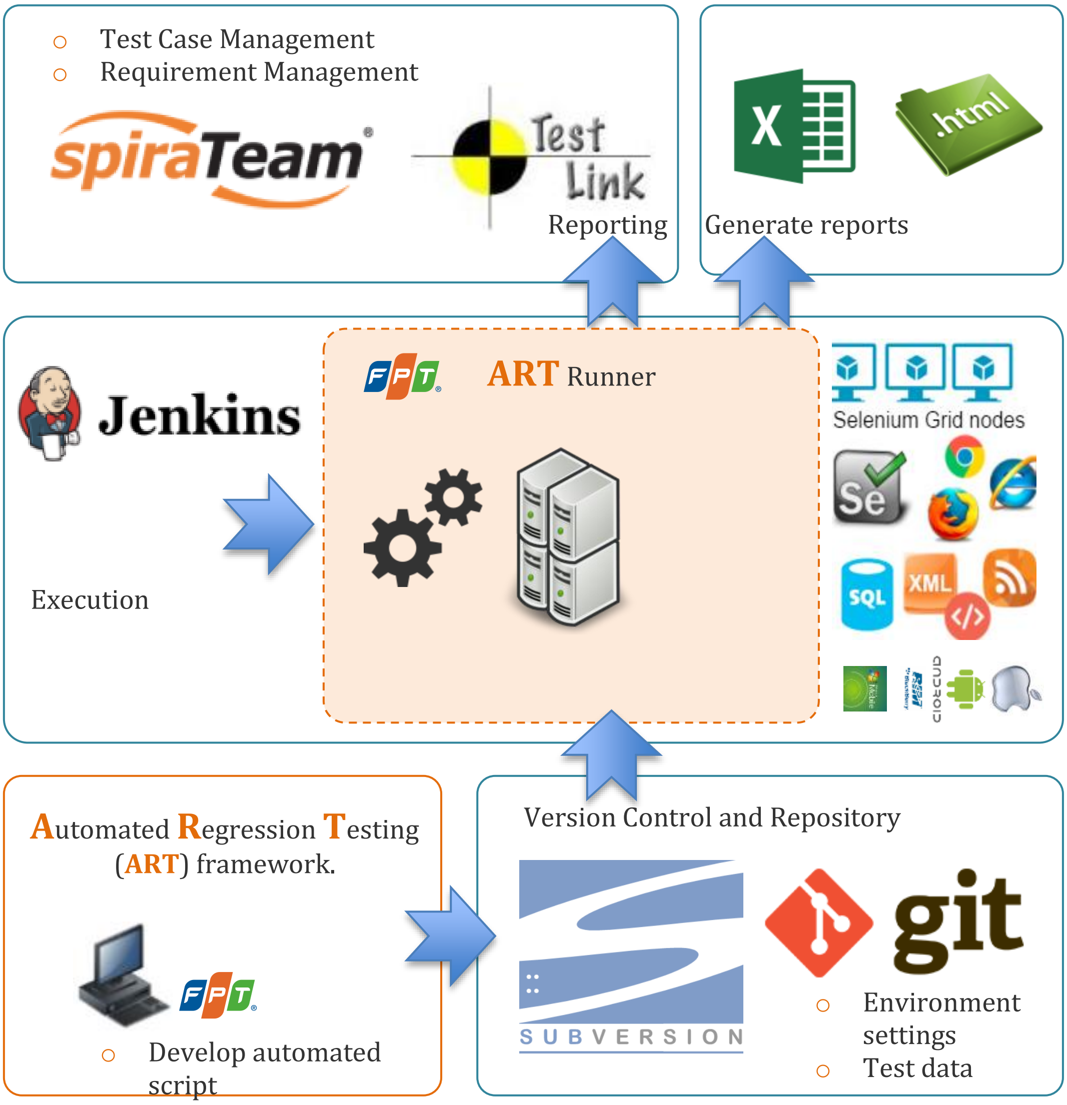
Agile testing practices require automation test integrated into CI which is built and executed on regular basis. The ART is not only designed in the way of immediate integration with CI but also provides script developer a comprehensive tool to create and execute test case at development environment.

5. ART Supported Environments

Application Type	Test Type	Application Under Test Description
Web Application	Functional	• Web application with UI running under browsers: Chrome, Firefox and IE.
Web Service	Functional	• SOAP and Restful web services
Mobile	Functional	• Android; iOS
Linux Shell	Functional	• SSH remote command line call and verification
HTTP Based Apps	Performance	• Web App and Web Service

6. ART Architecture Overview

6.1. Workflow in CI



- Test Case Management
- Requirement Management

Reporting

Generate reports

Execution

ART Runner

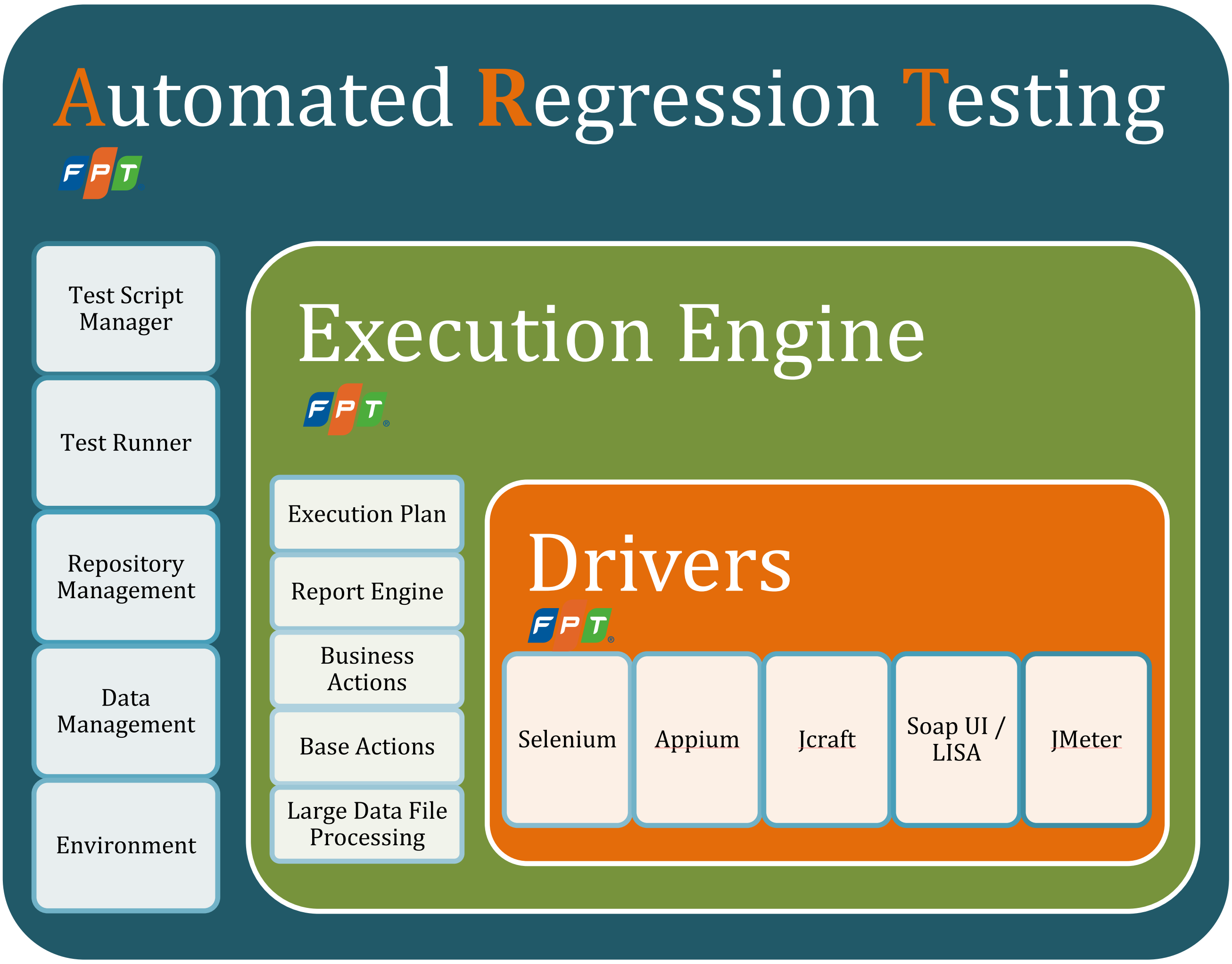
Automated Regression Testing (ART) framework.

- Develop automated script

Version Control and Repository

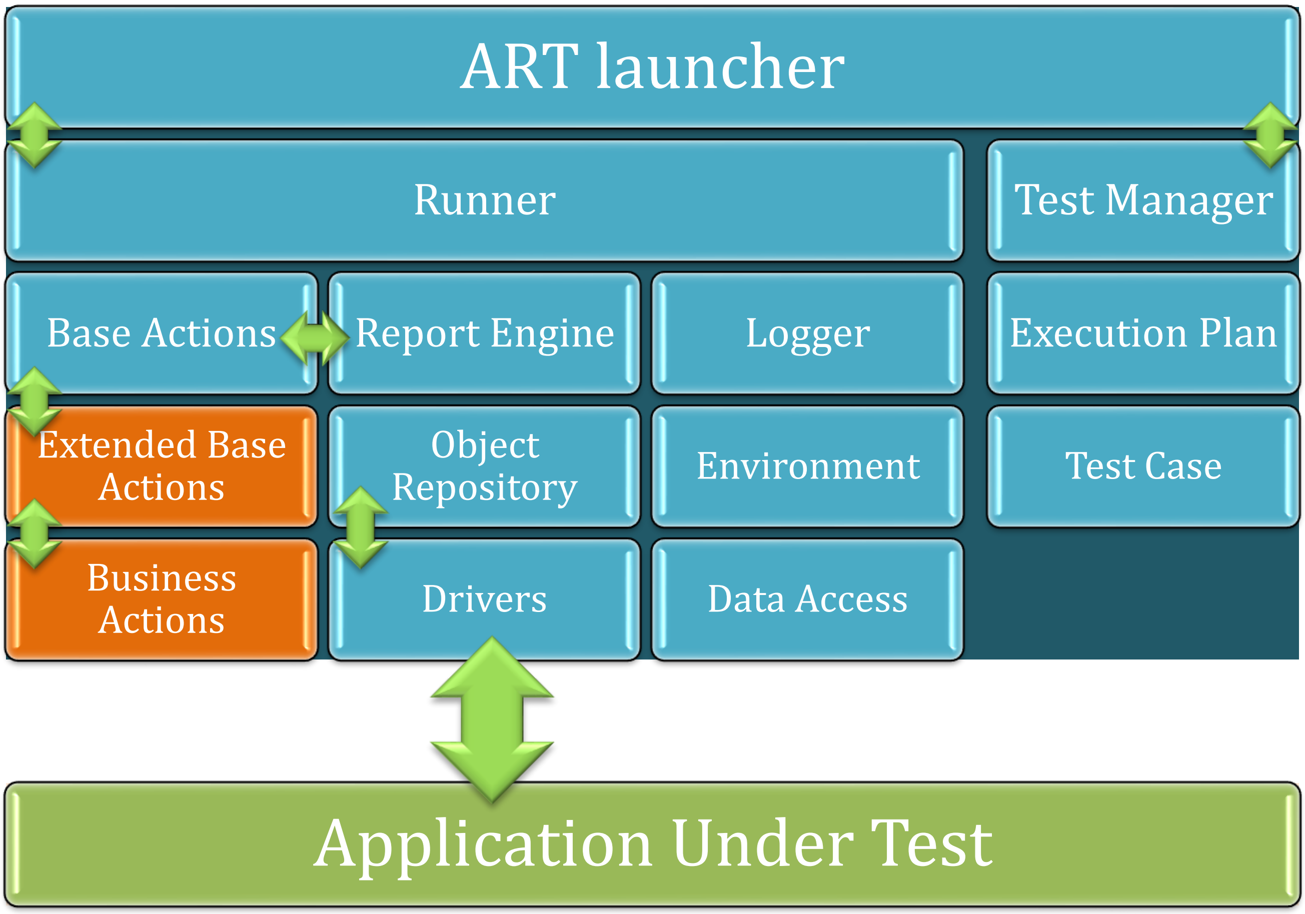
- Environment settings
- Test data

6.2. Component View



Powerful Framework
• Support multiple platforms from Web UI, Web Service, Linux Shell and Mobile. • Extendable actions for scripting. • Report engine uploads test result to SpiraTeam or TestLink by configured settings. • Hybrid framework to add new test case by adding new data set, or existing actions or keywords. • UI to manage execution plan for testing for each AUT released. • Distribute tests via Grid deployment.
Strong Collaboration
• Automated script uses test data created by Manual team. • Business workflow changes controlled by Manual team without re-writing test script.

6.3. Development Model



All components in the framework are designed flexibly so that they are generic patterns across projects. When the ART applies in one project, automated tester extends the base actions provided by the ART and implement business actions that project needs for automated testing.

Module	Description
ART Launcher	• This is entry point to decide whether Test Manage UI is opened for test management or a specified plan is executed immediately.
Test Manager	• UI to manage test execution plans and test cases.
Execution Plan	• Clone or update execution plan.
Test Case	• Create or update test case.
Logger	• Use the Log4j as a standard approach to log and diagnose problems in the detailed information about what occurs in a method at a granular level, additional with log4j you are able to control every aspect of the Art Framework through a simple configuration file. The Message Oriented-Report layer provides most of the functionality of common report messaging in the Art Framework.

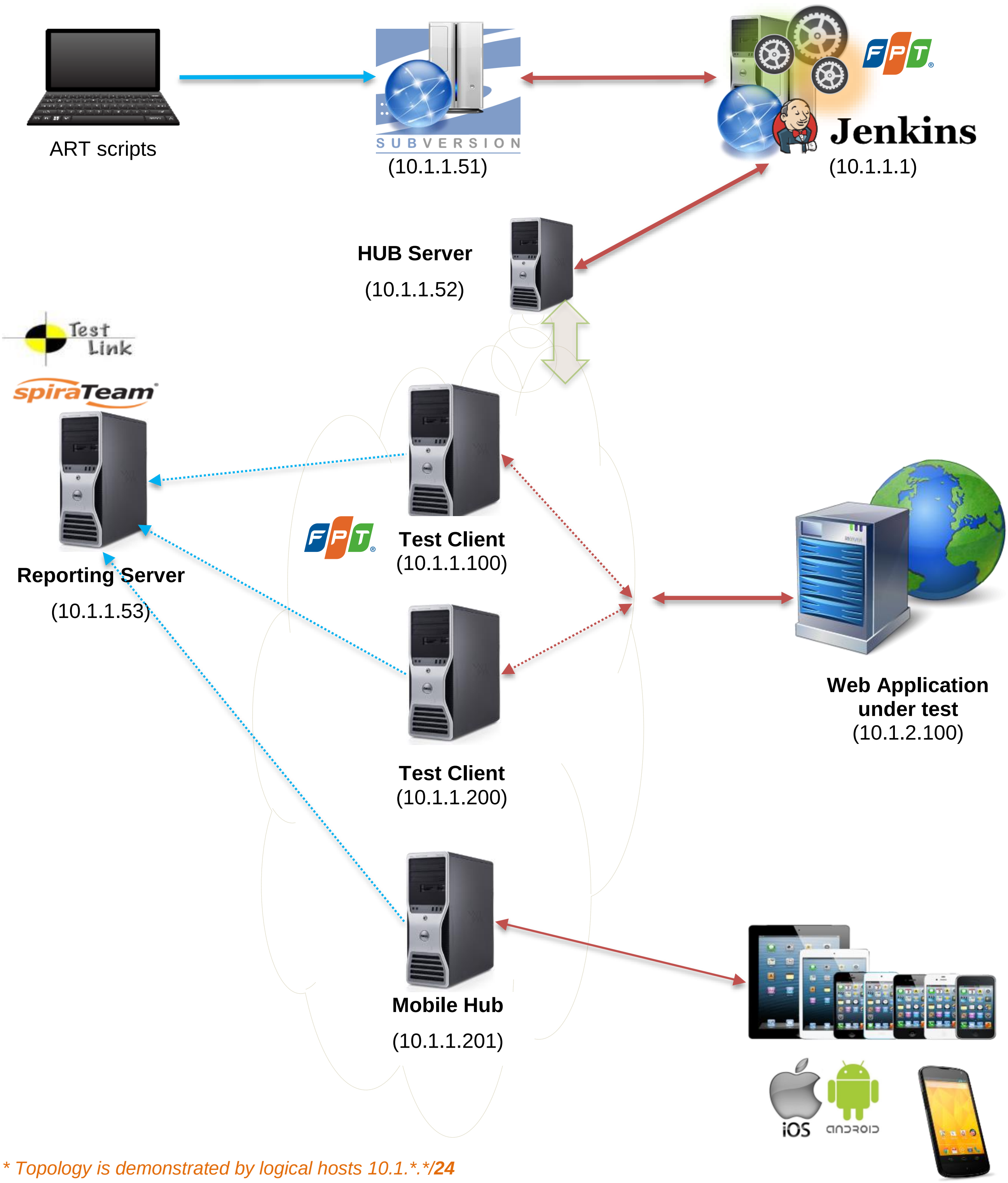
Module	Description
Environment	• Keep the Environment setup out of the code and pass them to the global Environment of the ART execution. They are properties in the Java platform with environment variables are key-value pairs.
Report Engine	• A consolidated report that gives you a diagrammatic representation of all test runs with passed tests and failed tests in the detailed manner and steps by steps via Excel Summary Report and also XML Summary Report. In case when the execution is complete, the ART framework is able to synchronize up the execution status and test results into the collaborative hubs such as SpiraTeam or TestLink.
Object Repository	• It is central object map which naming the object repository component, it is implemented as a collection of key-value pairs, with the keys as a logical name identify the objects and the value contain unique objects properties use to identify the object on the application under test.
Drivers	• Wrapper to communicate with different drivers to simulate end-user's activities on application under test.
Data Access	• The ART framework provides the common features that interacting with DB/SQL, Web Service/XML or Excel files to iterate different set of data with test cases automatically.
Base Actions	• It provides **powerful** keyword-driven test operations that let you perform various actions in your keyword-driven tests: performing common tasks, calling object methods, synchronizing object state and checking object states, and etc. • The ART designs two kinds of keyword: the **action** keywords enable to manipulate functional business events with detail reporting and the **assertion** keywords do verification as test case expected result.

Extended Base Actions	• The ART is designed in generic architecture so at project level you may want to customize and provide more functions on top of based actions. All you need to do is create a class to inherit the class BaseTest as following.
	`import z8.art.aut.BaseTest`
	`public class ProjectBaseTest extends BaseTest`

Business Actions	• This module is to implement business actions to test the UAT. The business actions will inherit or has all **base actions** provided by the ART and **extended base actions** provided at project level. • This is the only layer that user can use to design flow of testing, test case.

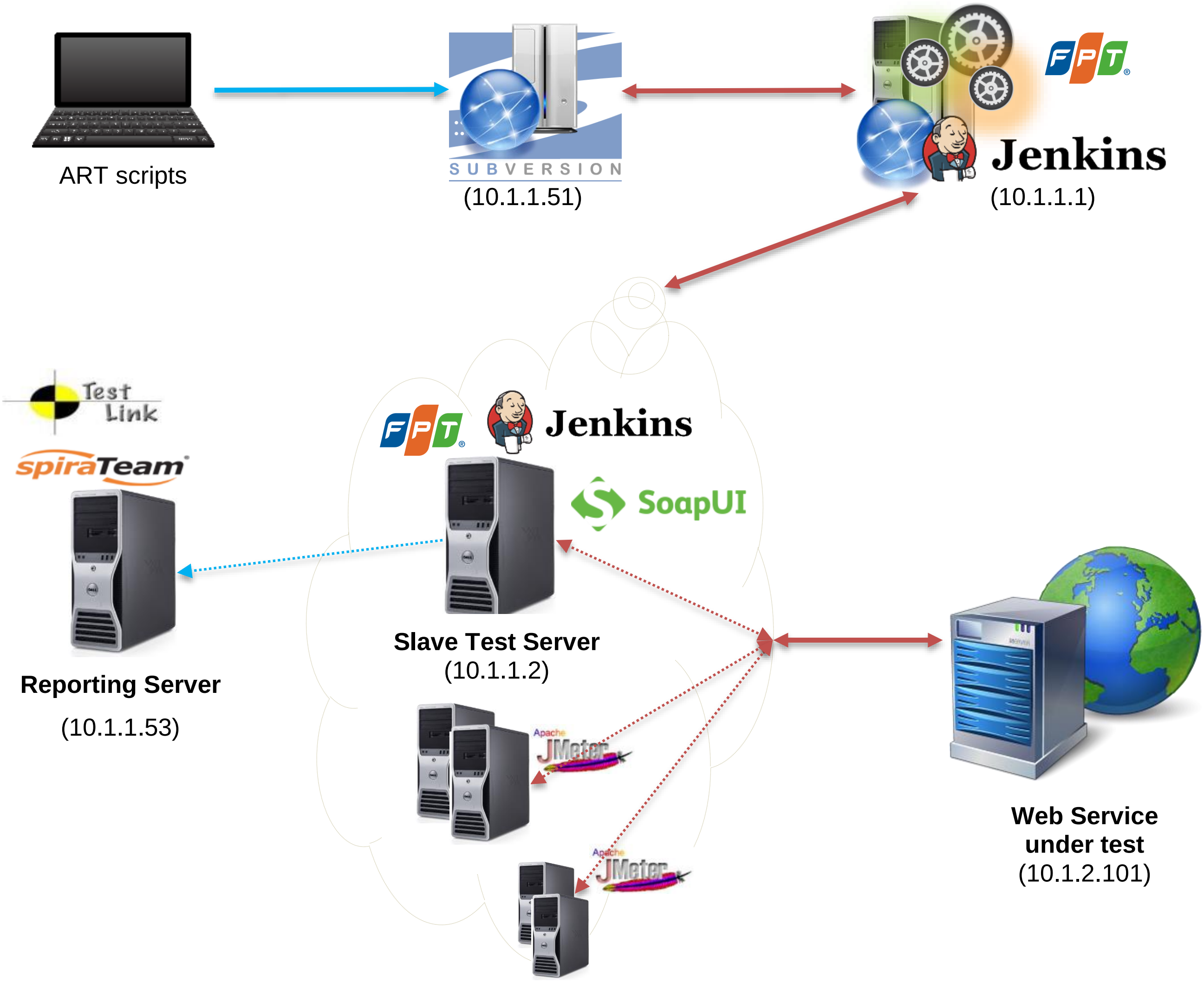
6.4. Types of Deployment

6.4.1. Deployment for Web UI and Mobile testing



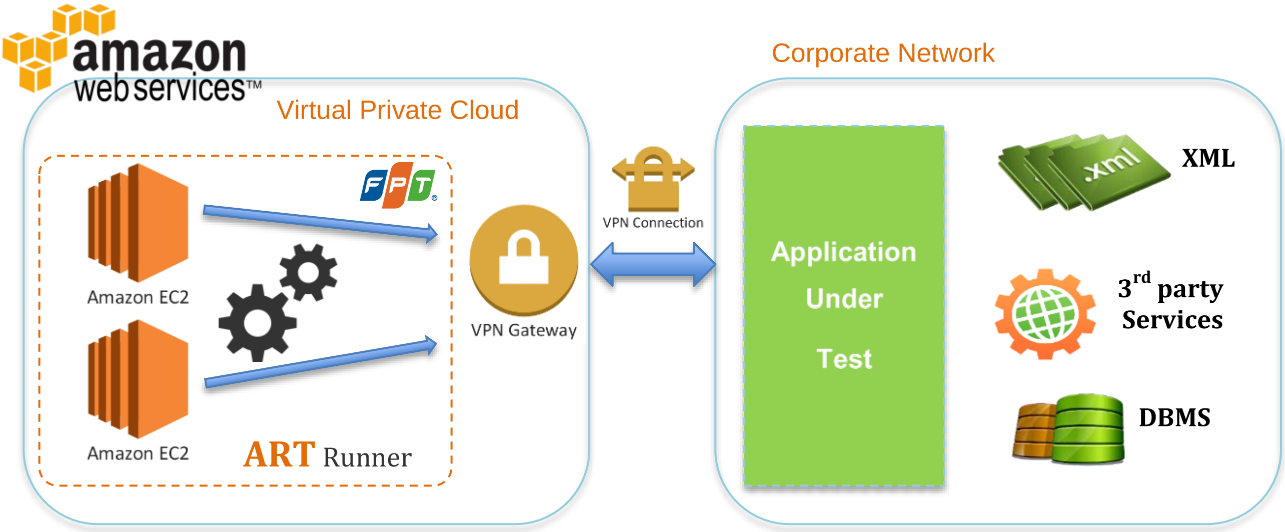
** Topology is demonstrated by logical hosts 10.1.*.*/**24***

6.4.2. Deployment for Web Service and Performance testing



** Topology is demonstrated by logical hosts 10.1.*.*/**24***

6.4.3. Deployment on Cloud (AWS)

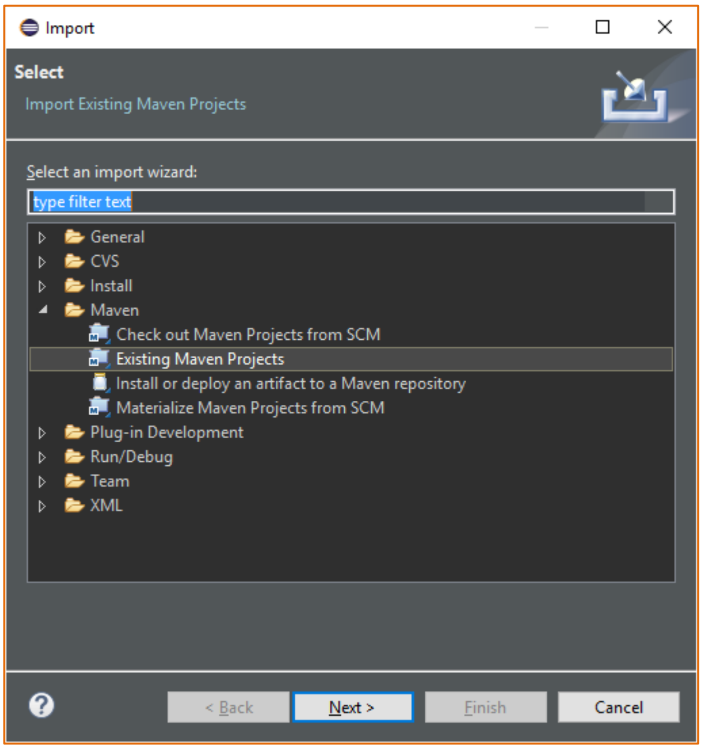


7. ART User Manual

7.1. Quick Start

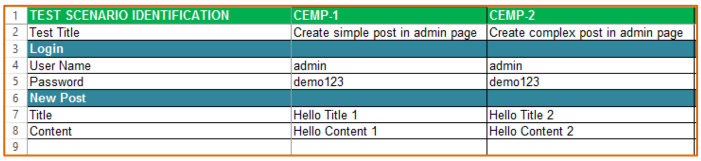
Step 1: Import Maven project from Eclipse

- File > Import…



Step 2: Prepare sample test data

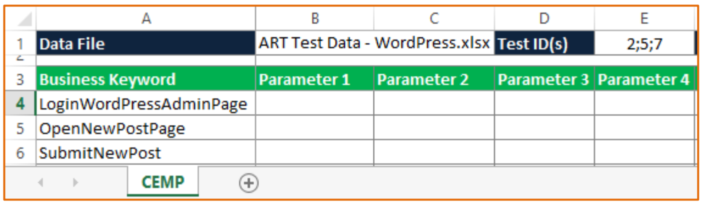
- **$data**ART Test Data - WordPress.xlsx
- Date set for ONE test case is populated by ONE column. It is looked up by test case ID. [BR-8]



1	TEST SCENARIO IDENTIFICATION	CEMP-1	CEMP-2
2	Test Title	Create simple post in admin page	Create complex post in admin page
3	Login		
4	User Name	admin	admin
5	Password	demo123	demo123
6	New Post		
7	Title	Hello Title 1	Hello Title 2
8	Content	Hello Content 1	Hello Content 2
9			

Step 3: Prepare test case flow

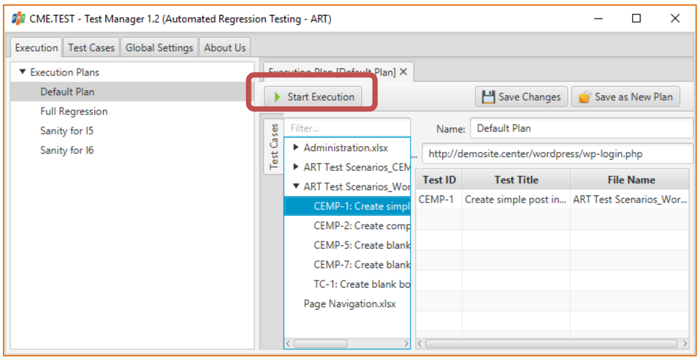
- **$scenarios**\ART Test Scenarios_WordPress.xlsx
- Data-driven test case ID(s) are generated by semicolon. [BR-7]



	A	B	C	D	E
1	Data File	ART Test Data - WordPress.xlsx		Test ID(s)	2;5;7
3	Business Keyword	Parameter 1	Parameter 2	Parameter 3	Parameter 4
4	LoginWordPressAdminPage				
5	OpenNewPostPage				
6	SubmitNewPost				

Step 4: Launch Test Manager

- Run *main()* method in class: **z8.art.Runner**
- Select **Default Plan** and click “Start Execution” button.



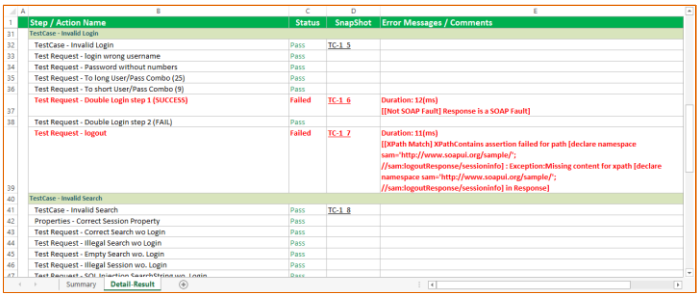
Name: Default Plan

http://demosite.center/wordpress/wp-login.php

Test ID	Test Title	File Name
CEMP-1	Create simple post in...	ART Test Scenarios_Wor...

Step 5: Review detail test result

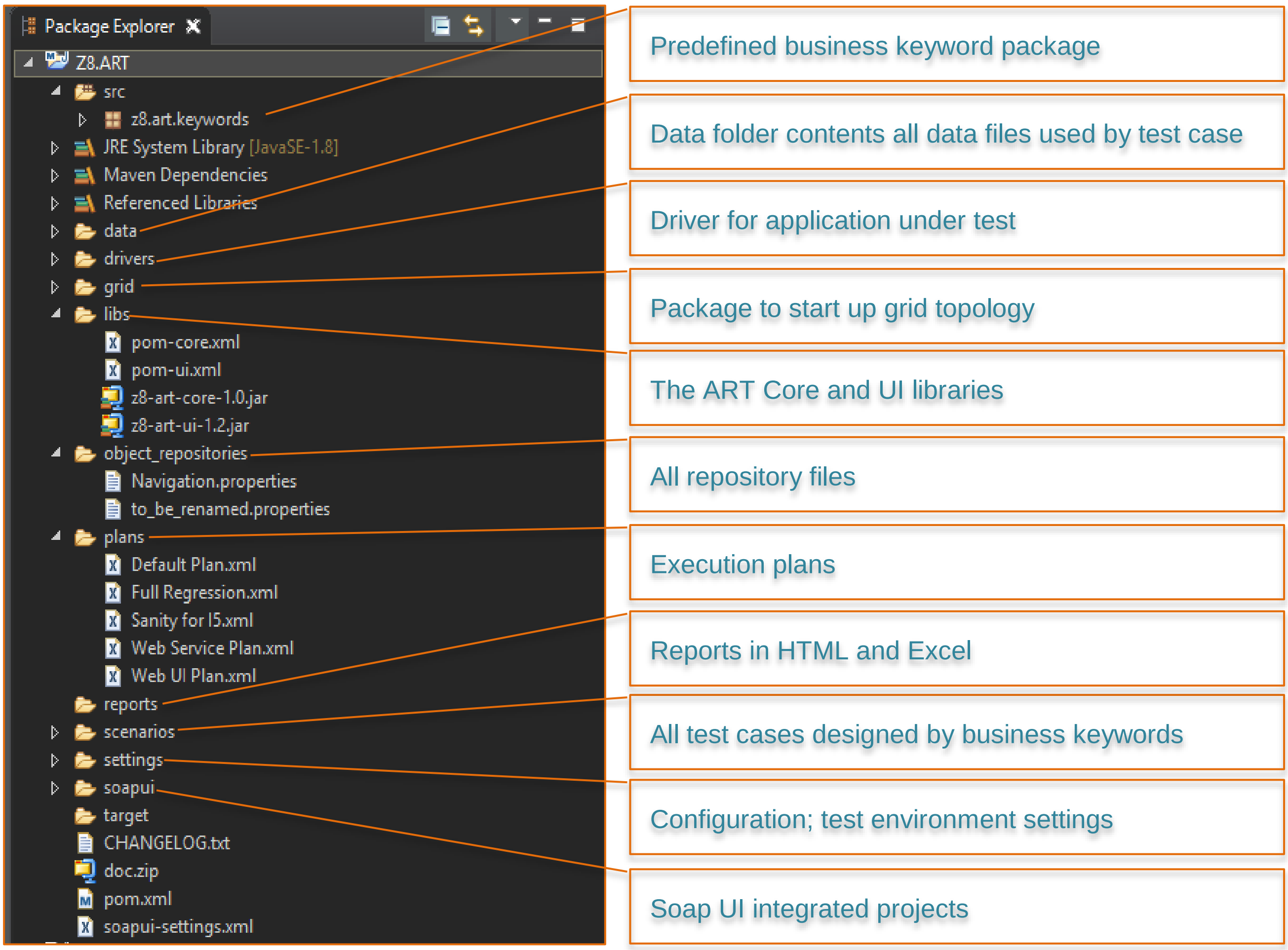
- The report as right depict allows to see comprehensive test step result which has screen shot and error messages for every failed step and at the end of each business action.
- Test result can be configured to be uploaded to SpiraTeam or TestLink on-the-fly.



	B	C	D	E
1	Step / Action Name	Status	SnapShot	Error Messages / Comments
31	TestCase - Invalid Login			
32	TestCase - Invalid Login	Pass	TC-1_5	
33	Test Request - login wrong username	Pass		
34	Test Request - Password without numbers	Pass		
35	Test Request - To long User/Pass Combo (25)	Pass		
36	Test Request - To short User/Pass Combo (9)	Pass		
37	Test Request - Double Login step 1 (SUCCESS)	Failed	TC-1_6	Duration: 12(ms) [[Not SOAP Fault] Response is a SOAP Fault]
38	Test Request - Double Login step 2 (FAIL)	Pass		
39	Test Request - logout	Failed	TC-1_7	Duration: 11(ms) [[XPath Match] XPathContains assertion failed for path [declare namespace sam="http://www.soapui.org/sample/"; //sam:logoutResponse/sessioninfo] : Exception:Missing content for xpath [declare namespace sam='http://www.soapui.org/sample/'; //sam:logoutResponse/sessioninfo] in Response]
40	TestCase - Invalid Search			
41	TestCase - Invalid Search	Pass	TC-1_8	
42	Properties - Correct Session Property	Pass		
43	Test Request - Correct Search wo Login	Pass		
44	Test Request - Illegal Search wo Login	Pass		
45	Test Request - Empty Search wo. Login	Pass		
46	Test Request - Illegal Session wo. Login	Pass		
47	Test Request - SQL Injection SearchString wo. Login	Pass		

7.2. General Functionalities

7.2.1. Package Structure



Predefined business keyword package

Data folder contents all data files used by test case

Driver for application under test

Package to start up grid topology

The ART Core and UI libraries

All repository files

Execution plans

Reports in HTML and Excel

All test cases designed by business keywords

Configuration; test environment settings

Soap UI integrated projects

[BR-1]. All classes in keyword package will be loaded automatically and their methods are also available to be called at UI layer.

[BR-2]. All file in repository folder will be loaded automatically and objects inside are available to be used by script.

7.2.2. Global Environment Setting

It is located at: $\settings\env.properties

Variable Name	Values / Default Value	Description
UAT_REMOTE_HUB_URL	[Blank]	Remote HUB URL; Blank (Default): Execute locally
UAT_BROWSER	Firefox; Chrome; IE; Android	Platform name to be tested
UAT_FIREFOX	[Optional]	Location of Firefox executable file

Variable Name	Values / Default Value	Description
UAT_OBJECT_TIMEOUT	20	Default object recognition timeout in seconds.
UAT_MOBILE_CAP_FILE	cap_Node1.properties	File name of Mobile Capabilities in folder drivers
REPORT_FILE_NAME	[Blank]	Provide fixed report file name # - Blank: Auto generate test report # - A file name for XLSX or HTML extension. i.e.: Report_ART, it will generate Report_ART.xlsx and Report_ART.html
REPORT_HTML		[True, False] setting whether the report HTML is generated or not
SPIRA_TEAM_ENABLED	TRUE; FALSE	Enable / Disable SpiraTeam integration.
SPIRA_TEAM_URL		Specify SpiraTeam URL
SPIRA_TEAM_USER_NAME		Specify SpiraTeam login username.
SPIRA_TEAM_PASSWORD		Specify SpiraTeam login password.
SPIRA_TEAM_PROJECT_ID		Specify SpiraTeam project id
SPIRA_TEAM_RUNNER		
SPIRA_TEAM_RUNNER_TEST		
SPIRA_TEAM_RELEASE_ID		The release id creates on the SpiraTeam
SPIRA_TEAM_TESTSET_ID		The test set id creates on the SpiraTeam
TESTLINK_ENABLED	TRUE; FALSE	Enable / Disable TestLink integration.
TESTLINK_XMLRPC_URL		
TESTLINK_DEV_KEY		
TESTLINK_TESTPLAN_ID		
TESTLINK_BUILD_ID		
TESTLINK_BUILD_NAME		
TESTLINK_PLATFORM_ID		
TESTLINK_PLATFORM_NAME		
SSH_HOST		The IP address or domain name that you are trying to connect to
SSH_PORT		Port to connect to on the remote host
SSH_USER		The user to log in as on the remote machine

Variable Name	Values / Default Value	Description
SSH_PASSWD		The password to log in as SSH_USER on the remote machine

7.2.3. Entry point of ART Launcher

By default the ART launcher will open the Test Manager UI so that user can manage test execution plan and test cases.

In addition to that if user provides arguments to the launcher it will execute provided plan immediately instead of opening the Test Manager UI. The following is syntax and a list of supported arguments.

Syntax: java -jar **myproject.jar** *[arg 1] [arg value 1]…*

Argument Name	Required?	Description
-planFile	Yes	File name of execution plan which is required to run testing. The file is an XML file which is located in the folder $**plans\**
-envFile	No	• The environment file which describes environment under test. It is the properties file located in the folder $**settings\** • Default file name: $**settings\\env.properties**
-reportFile	No	• File name for reporting. • Default file name: it will be generated by date time convention: Report_<yyyy_mm_dd_hh_mm_ss>.xlsx

7.2.4. How Data-driven works

Following are key objectives to apply **data-driven scripting** into the ART.

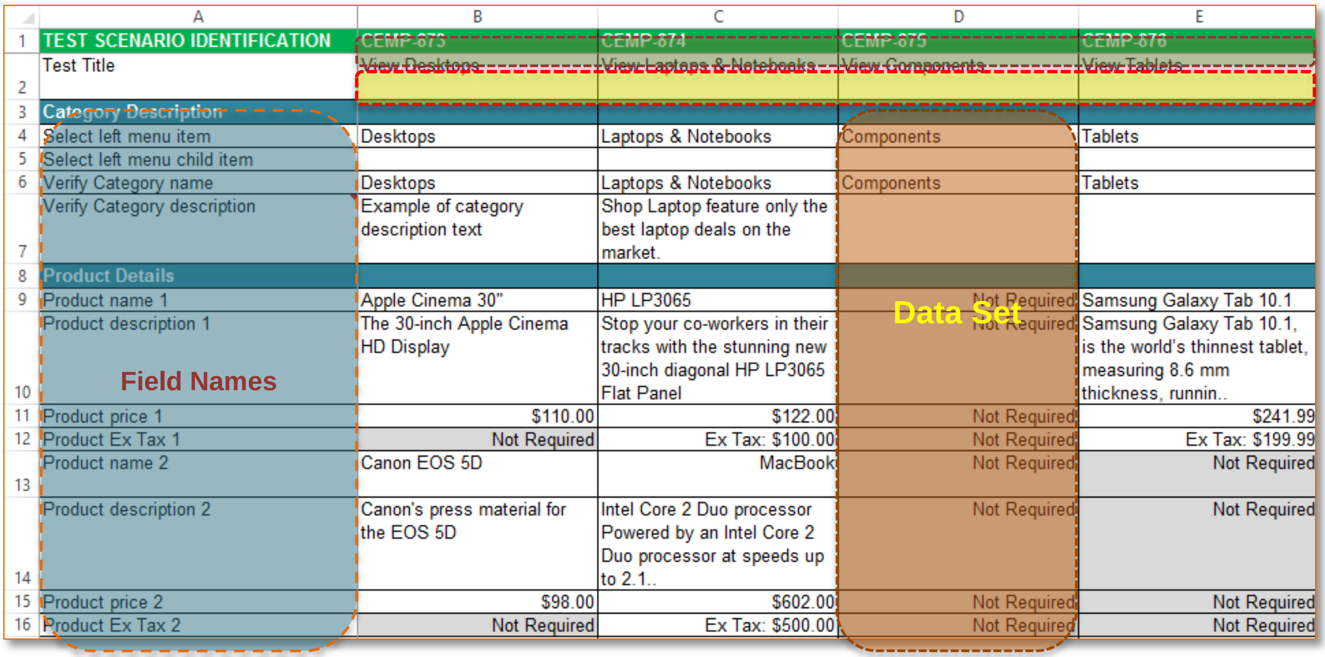
- **Separate test data** out of test script for easy maintenance.
- Create **new test case** by creating **new set of data** and do not need to create additional script.

The data are standardized by following rules:

[BR-DATA-1]. One data set is values in a column starting from B. The ART framework identifies data set by **Test Case ID** at the **first row**. For example: test case ID: CEMP-871

[BR-DATA-2]. "**Test Title**" at 2nd row is test case title which is used when reporting.

[BR-DATA-3]. Field Name is term which is mentioned in the document and scripting data indicates column A from 3rd row downward.



	A	B	C	D	E
1	TEST SCENARIO IDENTIFICATION	CEMP-873	CEMP-874	CEMP-875	CEMP-876
2	Test Title	View Desktops	View Laptops & Notebooks	View Components	View Tablets
3	Category Description				
4	Select left menu item	Desktops	Laptops & Notebooks	Components	Tablets
5	Select left menu child item				
6	Verify Category name	Desktops	Laptops & Notebooks	Components	Tablets
7	Verify Category description	Example of category description text	Shop Laptop feature only the best laptop deals on the market.		
8	Product Details				
9	Product name 1	Apple Cinema 30"	HP LP3065	Not Required	Samsung Galaxy Tab 10.1
10	Product description 1	The 30-inch Apple Cinema HD Display	Stop your co-workers in their tracks with the stunning new 30-inch diagonal HP LP3065 Flat Panel	Not Required	Samsung Galaxy Tab 10.1, is the world's thinnest tablet, measuring 8.6 mm thickness, runnin..
11	Product price 1	$110.00	$122.00	Not Required	$241.99
12	Product Ex Tax 1	Not Required	Ex Tax: $100.00	Not Required	Ex Tax: $199.99
13	Product name 2	Canon EOS 5D	MacBook	Not Required	Not Required
14	Product description 2	Canon's press material for the EOS 5D	Intel Core 2 Duo processor Powered by an Intel Core 2 Duo processor at speeds up to 2.1..	Not Required	Not Required
15	Product price 2	$98.00	$602.00	Not Required	Not Required
16	Product Ex Tax 2	Not Required	Ex Tax: $500.00	Not Required	Not Required

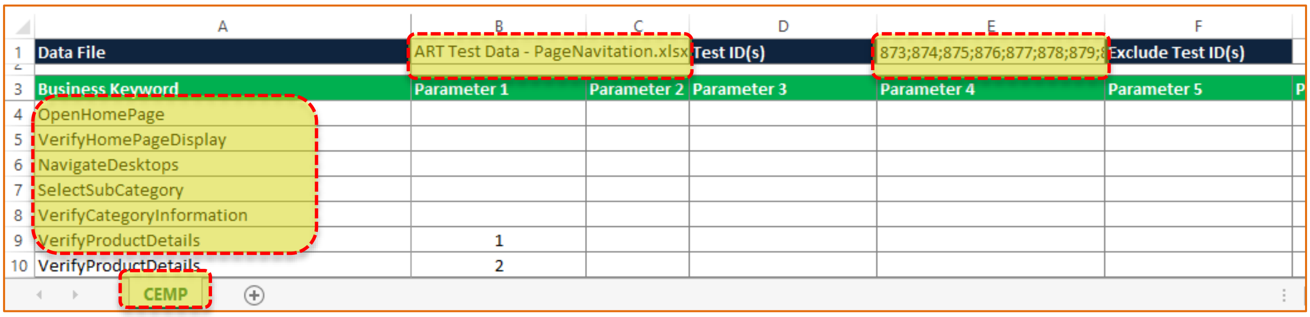
The data is stored in Excel format (prefer *.**xlsx**). By using Excel format, we can leverage the powerful features when designing test data such as: formula for calculation, formatting and etc.

7.2.5. How Keyword-driven works

Following are key objectives to apply **keyword-driven scripting** into the ART.

- **Business user** can **create new test case** using **existing business keywords**.
- Business user can **drive test flow without having programming knowledge.**

Keywords or actions in the ART are stored in Excel files under **$scenarios*.xlsx**. The execution performs keywords from top to bottom till a blank cell in the column **A**.



	A	B	C	D	E	F
1	Data File	ART Test Data - PageNavitation.xlsx		Test ID(s)	873;874;875;876;877;878;879;8	Exclude Test ID(s)
3	Business Keyword	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
4	OpenHomePage					
5	VerifyHomePageDisplay					
6	NavigateDesktops					
7	SelectSubCategory					
8	VerifyCategoryInformation					
9	VerifyProductDetails	1				
10	VerifyProductDetails	2				

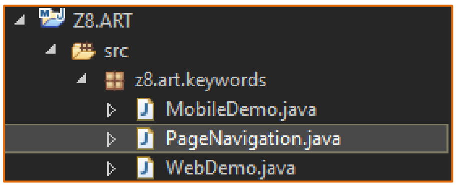
The file use the column **B1** to link to the data file which is used by all test cases in this. *Refer [BR-7]*

The ART is considered as hybrid framework and it generates test cases having the same test flow by spreadsheet name and test case IDs separated by semicolon (;). *Refer [BR-8]*

7.2.6. Use test data to drive test flow

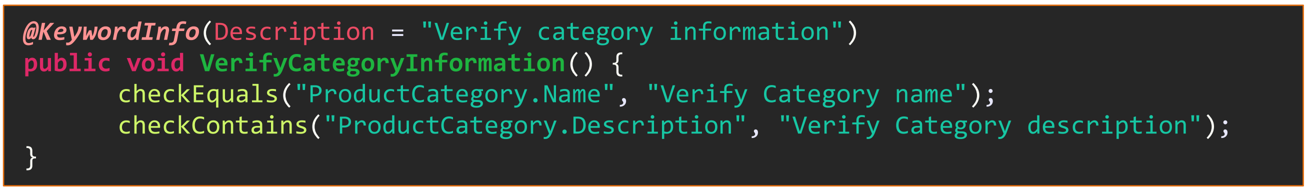
7.2.7. How to write a keyword / action

A keyword / action must be written in a class under package name: **z8.art.keywords**. A class should be named as functional area of the UAT.



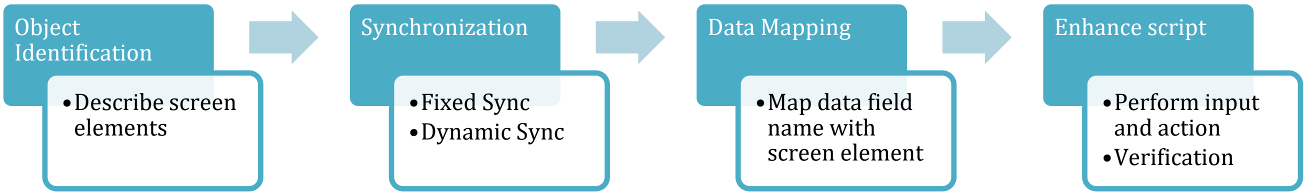
```
Z8.ART
  src
    z8.art.keywords
      MobileDemo.java
      PageNavigation.java
      WebDemo.java
```

A keyword must have the annotation ***@KeywordInfo*** with **Description**.



```
@KeywordInfo(Description = "Verify category information")
public void VerifyCategoryInformation() {
      checkEquals("ProductCategory.Name", "Verify Category name");
      checkContains("ProductCategory.Description", "Verify Category description");
}
```

7.2.8. Scripting Process and Base Action Convention



Object Identification
- Describe screen elements

Synchronization
- Fixed Sync
- Dynamic Sync

Data Mapping
- Map data field name with screen element

Enhance script
- Perform input and action
- Verification

Base Action Prefix	Example	Description
wait	wait waitExist waitNotExist	• **Synchronization** action which allows script to be synchronized in fixed duration or dynamic duration with timeout. • Dynamic synchronization point will be passed when expected event appeared or timeout reached.
check	checkContains checkDisable checkEquals	• **Check or verification point** which is used to verify the actual and **expected result** in test case.
click	click checkHover	• **Perform action** again element on the UAT to simulate flow of test case.
type **set** **select**	typeText setText selectCheckBox	• **Perform input data** into element or select value of element on the UAT.
soapui	soapuiOpenProject soapuiExecuteProject	• **Soap UI integration** action which is used to execute existing Soap UI project, test suite or test case.

Base Action Prefix	Example	Description
ssh	sshAsyncCmd sshExpect	- **SSH integration** action to call remote command lines executed on UNIX based OS and verified expected output.
jmeter	TBD	- **JMeter integration.**

7.2.9. Reporting

Currently the ART provides integration to upload test result to SpiraTeam and TestLink web based tools. It also provides other types of report in Excel and HTML format. The settings in $settings\env.properties allow to turn on/off the report integration or report format types.

In order to provide convenient for scripting the ART framework provides following functionalities while reporting.

[BR-3]. A screenshot is captured and associated with report step at the end of each keyword.

[BR-4]. A screenshot is captured automatically if the test step is failed.

[BR-5]. We SHOULD give indication, by based action `reportSnapShot()`, to capture screenshot for the next test step as test evident.

[BR-6]. Test case execution result is uploaded to test case management system after every test case run completely. This is to eliminate the report tracking lost when unexpected crash happens.

7.2.10. ART integrated into Jenkins

The ART is built using Maven so the Jenkins is required to install plug-in Maven. The ART is separated into two components which include UI and Core. The Jenkins job is configured to deploy two components as JAR files to Maven local repository. This is completed process to configure Maven job:

#	Maven Goal	Parameters	Description
1	**install:install**	-file=libs/z8-art-core-1.0.jar -pomFile=libs/pom-core.xml	- Deploy core to local Maven repository
2	**install:install**	-file=libs/z8-art-ui-1.2.jar -pomFile=libs/pom-ui.xml	- Deploy UI to local Maven repository
3	**clean install**		- Build current project using the ART

The final step is to launch the ART execution plan. For more detail refer to ***7.2.3 Entry point of ART Launcher***.

7.2.11. Parallel Test Execution

TBD

7.3. Business Rules

[BR-7]. Test case ID (s) will be generated based on sheet name and Test ID(s) value (Cell: C1). Sheet Name is prefix and Test ID(s) are separated by semicolon;

For example:

- Sheet Name: CEMP
- Test ID (s): 1;2;5

➢ Test case ID (s) generated: CEMP-1; CEMP-2; CEMP-5

[BR-8]. Cell B1: data file name will be used by test case to look up data. The file name is located in the folder $data\

7.4. Exposure Consideration for Base or Business Actions

TBD

8. Releases

8.1. Baseline on May 1, 2015

ART Framework	•ART Core 1.0 •ART UI 1.2
Java 8	•JDK 8 (u51)+ •Java FX
Selenium	•Selenium 2.53.0 •Selenium server-standalone 2.53.1 (Grid)
Appium	•Appium client 4.0.0
Soap UI	•Soap UI 5.0; Soap UI 5.2.1 (Free edition - 7/2/2015)
Jcraft	•Jcraft 0.1.53
JMeter	•JMeter 3.0

8.2. Baseline on Dec 15, 2014

ART Framework	• ART 1.0
Java 7	• JDK 7
Selenium	• Selenium 2.44.0
Appium	• Appium client 4.0.0
Jcraft	• Jcraft 0.1.53

9. Case Studies

Project 1

- Size: **30 MM**
- Duration: 6 months

Testing:

- Web Service (SOAP)

ROI: 12 MM

Test Case

- Total: 1,250

Automated: 750 TCs

- Executed cycles: 20

Project 2

- Size: **99 MM**
- Duration: 9 months

Testing: Web UI / Web Service (SOAP/Restful)

ROI: 20.8 MM

Test Case

- Total: 4,492

Automated: 2,683 TCs

- Executed cycles: 26

Project 3

- Size: **64 MM**
- Duration: 8 months

Testing: Web UI / Linux Shell

ROI: 16 MM

Test Case

- Total: 345

Automated: **255** TCs

- Executed cycles: 28

Project 4

- Size: **154 MM**
- Duration: 7 months

Testing: Linux Shell / Large Data File Processing

ROI: 26.2 MM

Test Case

- Total: 2,249

Automated: 720 TCs

- Executed cycles: 26

10. Open to FPT and Support Team

The ART is generic and proven practical framework which has been applied in CME.TESHCM. It is separated from client's IP so it is open to FPT automation test community to apply or trained by FPT Software Corporate Training Centre. Following contact points are contactable for further support or request for the ART package.

Name	Role	Email
Triet To	Sr. Test Architect	TrietTX@fsoft.com.vn
Hoang Nguyen	Sr. Automation Test Consultant	HoangNV5@fsoft.com.vn
Phi Hoang	Product Owner	PhiH@fsoft.com.vn

11. References

[1]. Rex Black and Jamie L. Mitchell, "*Advanced Software Testing Vol. 3 Guide to the ISTQB Advanced Certification as an Advanced Technical Test Analyst*", 2011.