



Optimus

MASTERWORKS AUTOMATION FRAMEWORK

Overview

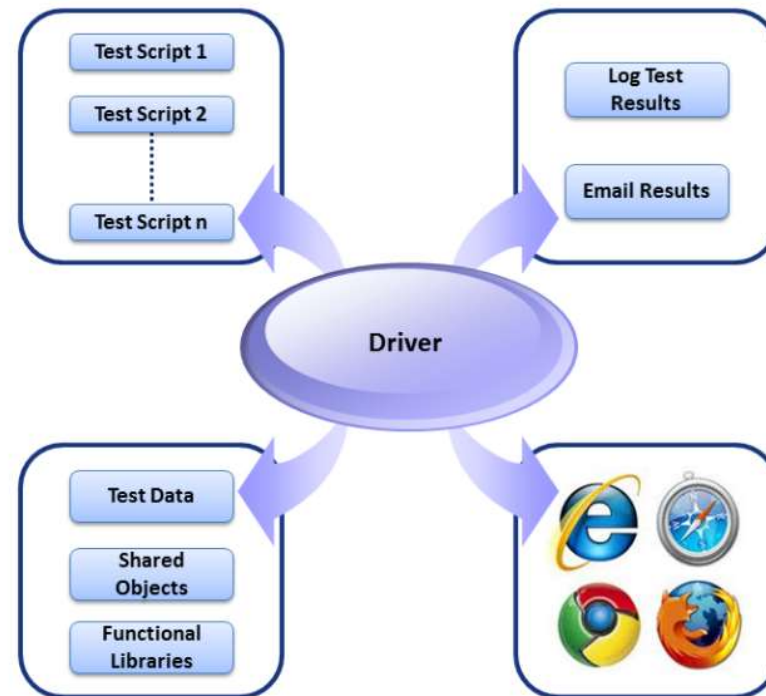
- ▶ About Automation Framework
- ▶ Architecture of Optimus
- ▶ Walkthrough
 - ▶ Adding a new test case in product and executing it
 - ▶ Setting up a new project
 - ▶ Executing test case from the new project
 - ▶ Naming conventions and test data files
 - ▶ Dependencies of Tests and Modules
 - ▶ Modifying the configurations
 - ▶ The App.config File
- ▶ The Road Map
- ▶ Questions

What is a Test Automation Framework?



Automation Framework

- ▶ Executing Test Scripts
- ▶ Managing Test Data
- ▶ Providing Shared Objects
- ▶ Executing Tests on Different Browsers
- ▶ Log Test Results
- ▶ Emails



Framework Architecture

Selenium

TEXT UTILS

CREATE NEW WEBSITE

DATABASE

GLOBAL VARIABLES

NAVIGATION

SHARED STEPS

EMAIL REPORTING

IMPORT LIBRARY DATA

UI CONTROLS

PUBLISH TO DASHBOARD

GENERIC FUNCTIONS

SCREEN RECORDING

SHARED STEPS

APP SERVER

XML HELPER

GENERIC FORM HANDLERS

MWPLATFORM

APPLICATION
MODULES

TEST CASES

PRODUCT
TEST BED

AUTOMATION START

PRODUCT AREAS

TEST SETTINGS

TEST DATA

Teams

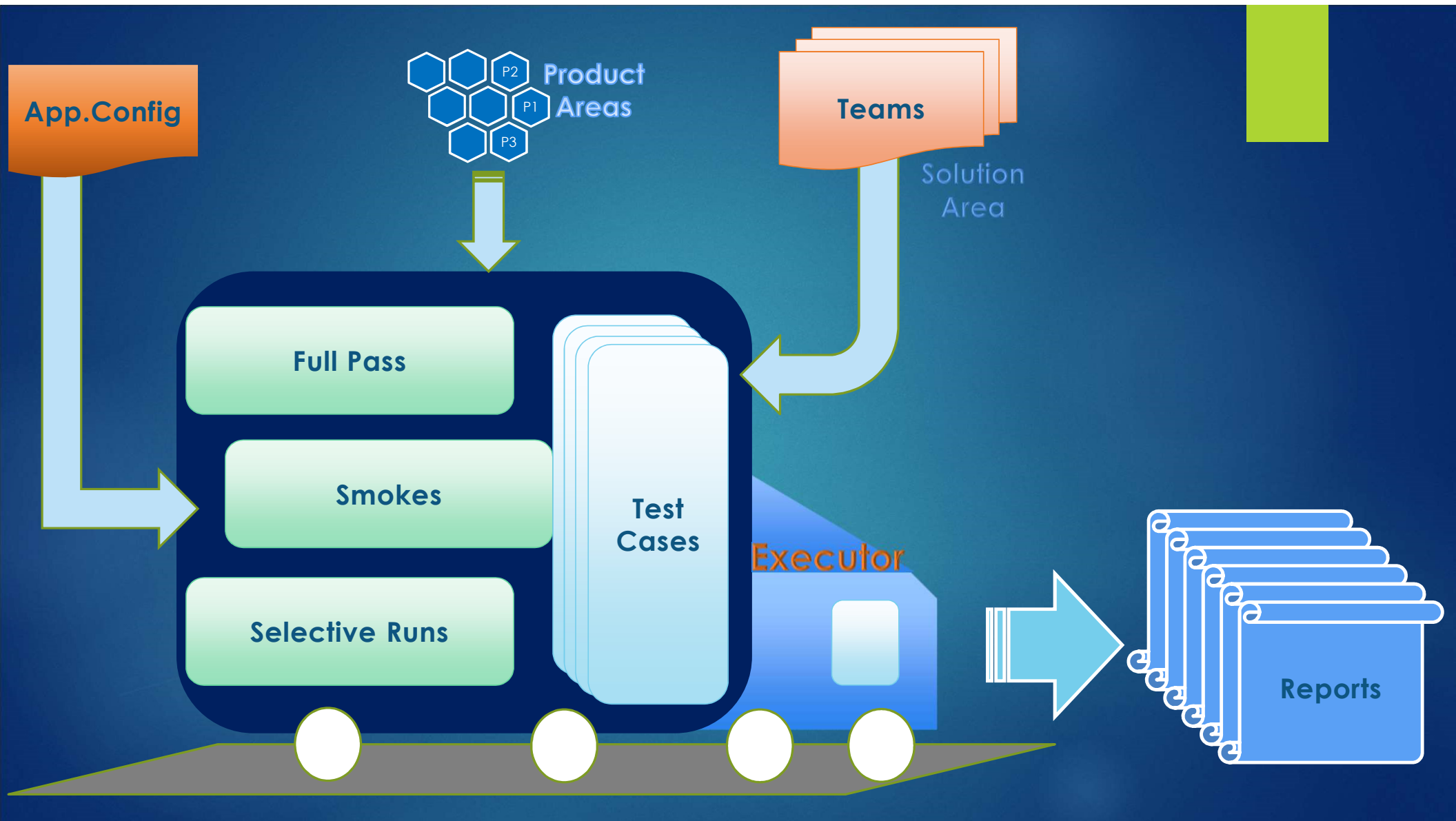
LIBRARY DATA

Configurations

APP.Config

Project
Environment Setup

Executor



Setting up a new TestBed

Create

- Create a new project

Configurations

- Post build event to copy the configurations to AutomationStart bin

DLL's and PDB's

- Post build event to copy all the dll's and pdb's to automation start bin folder

Declare the Team

- Add the Team Entry in Teams File

Default Team

- Specify the default Team in App.Config

GuideLines / Conventions

01

Naming of Test
case class and
Method

02

Naming of the
test data files

03

Namespace
under which all
the tests are
written

04

All the test
settings files
should be placed
in the test settings
folder

05

All the test data
files to be placed
in the test data
folder

Creating a new TestCase

1

Add the test case class in the test bed

2

Check if the Suite that it belongs to exists in the ProductAreas

3

Add the entry of the test case in the Test Settings file

Executing a test case

1

Command
Line
arguments

2

Check for the
suite

3

Check for the
executeVal
attribute

4

Check for the
Team that the
test belongs to

Changing the Settings

Configurations	Emails, Build URL, AppServer, DBServer, Publish to Dashboard, Upload modules, etc
Users	Define the Users that will be used for Automation
Teams (TestFolderPaths)	Specify the Test DLL's and assemblies
App.Config	Default Team, Create ScratchBuild, ReportType, ModuleUploadThroughAPI, UseGenericLogging, Priorities.

Dependencies of Tests and Modules

```
<Suite id="7" planningmod="yes">Planning-Planned Projects.xml</Suite>  
<Suite id="8" planningmod="yes">Program.xml</Suite>  
<Suite id="9">Project Scoring.xml</Suite>  
<Suite id="10">Project Submittals.xml</Suite>  
<Suite id="11">Documents.xml</Suite>  
<Suite id="12">Project Fund Management.xml</Suite>  
<Suite id="13">Budget Management.xml</Suite>  
<Suite id="14" DependentOnModules="7">bidmanagement.xml</Suite>  
<Suite id="15" DependentOnModules="7,14">Configuration.xml</Suite>
```

```
<test executeVal="Yes">TC_117383_CreateNewSkies</test>  
<test executeVal="Yes" DependentOnTests="TC_117383_CreateNewSkies">TC_117384_ViewSkies</test>
```

Work In Progress

01

Parallel
Execution of
Modules and
tests

02

Generic Form
Handlers

03

API Testing