

THE MODERN SOFTWARE FACTORY



Model based Tests Execution/ Automation using Taurus

Sept 5th, 2017

Sudhrity Mondal, Director Presales

Agenda

- Model Based Testing using CA Agile Requirements Designer
- Test Execution Automation using CA BlazeMeter & Taurus
- Demonstration
- Q&A

How do you build better apps faster?

CA Agile Requirements Designer

Requirements : Lack of Clarity and Vision during Development & Test

USER



**BUSINESS
ANALYST**



PROGRAMMER



TESTER



The user knows what they want

The programmer writes the code

The analyst specifies what that is

The tester tests the code

The further the visions diverge, the greater the problems downstream

Requirements : Clarity and Vision during Development & Test

USER



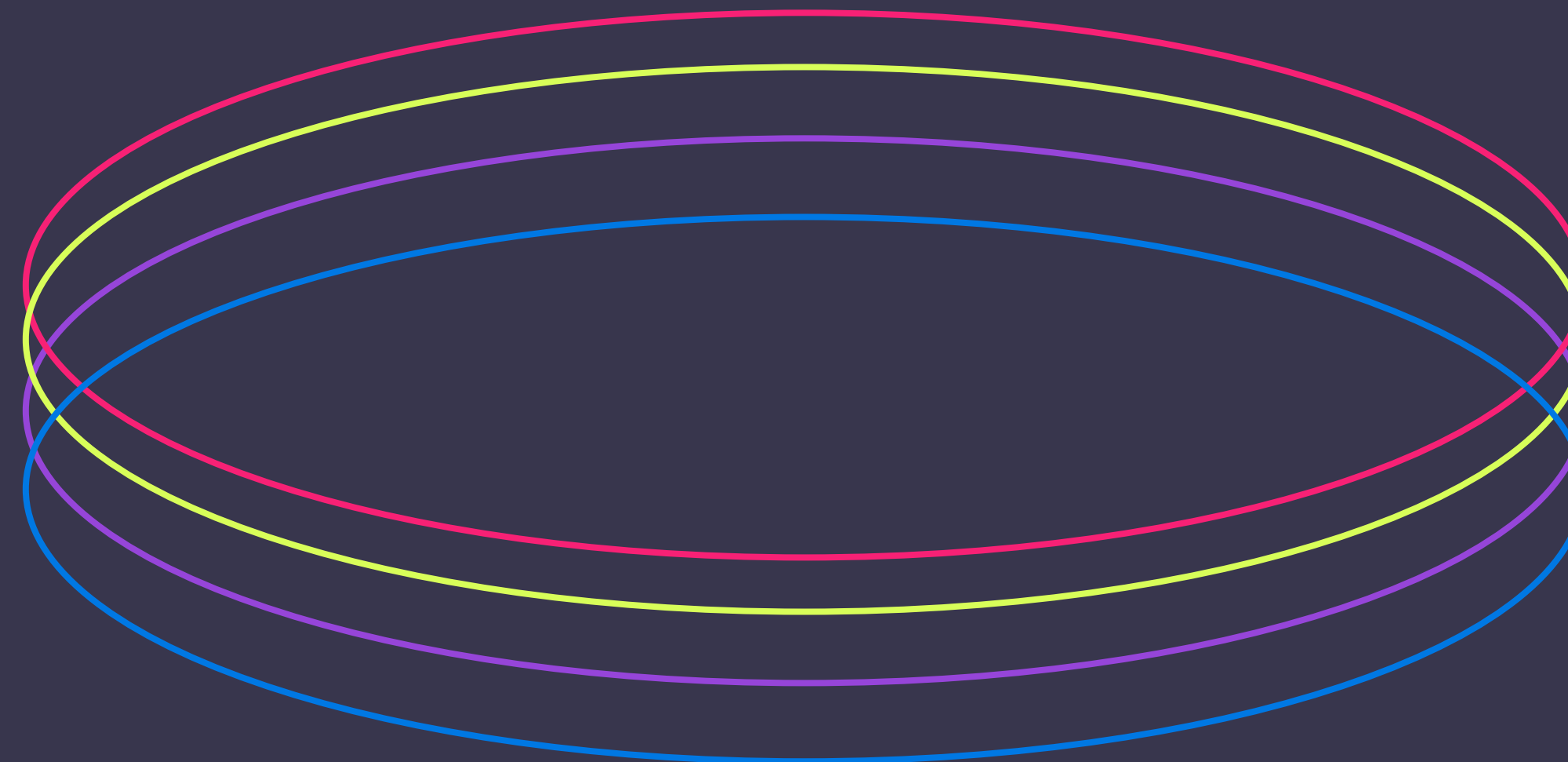
**BUSINESS
ANALYST**



PROGRAMMER



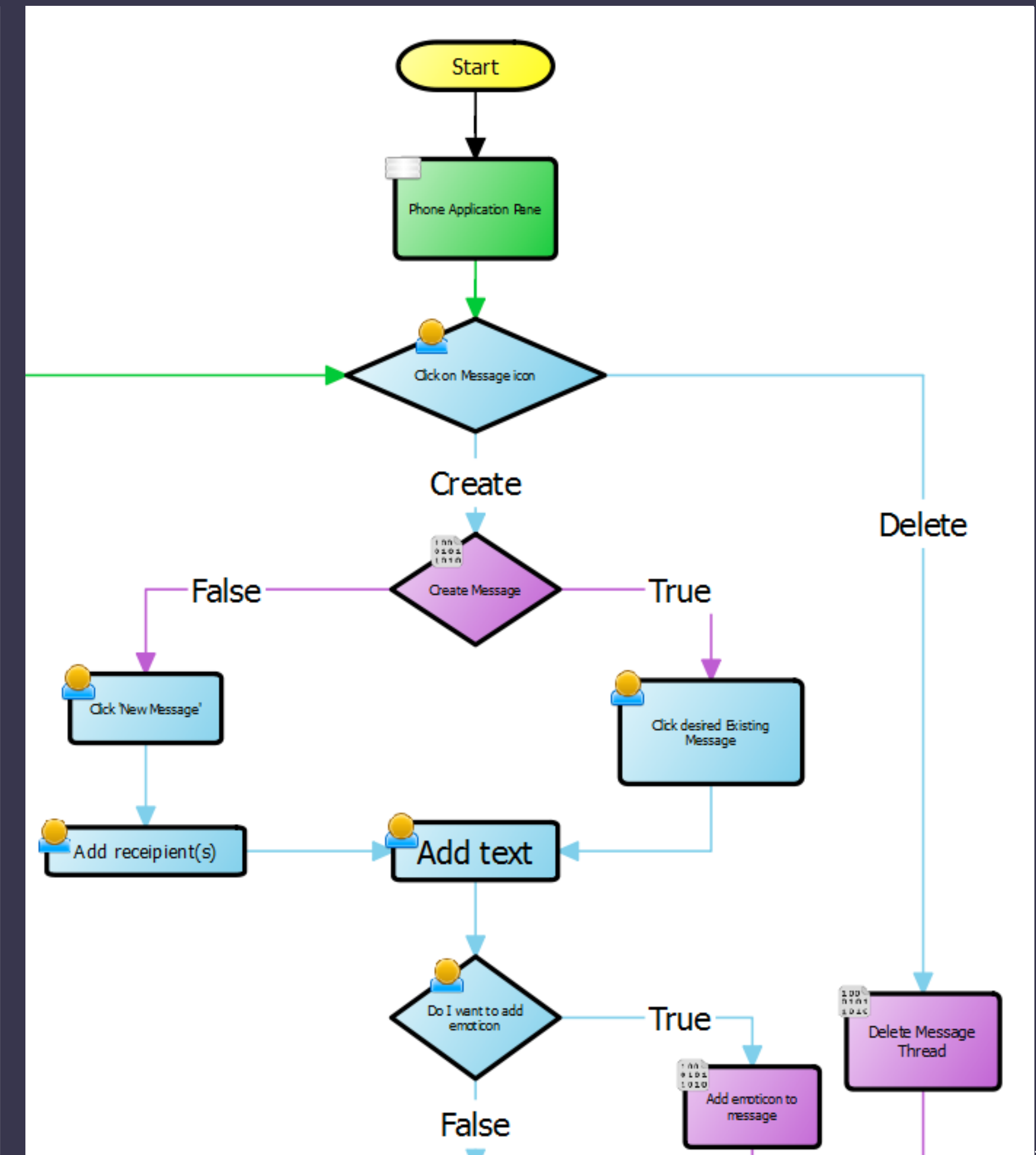
TESTER



**The more aligned the vision, the higher the quality app.
Fewer bugs and faster product delivery.**

Improve Quality & Efficiency using Model Based Testing

- **Build better requirements**, improve communication, visual flows to validate requirements.
- **Automate Creation and Optimization of Test Cases**, maximum coverage with minimum tests cases.
- **Identify data requirements** supporting Test Cases, both positive and negative testing.
- **Understand/document Test Coverage & Risk**.
- Automatically **identify the impact of requirement changes** to Test Cases maintenance.
- Supports modeling and testing at all SDLC levels, unit, functional, integration, regression...
- **Integration with testing frameworks**, create/maintain Test Automation scripts.
- Integration with existing Agile Management & Testing tools.



Layered approach to Model Based Testing

Overlaying all Layers

Model <ul style="list-style-type: none">▪ Business Flows based on User Stories▪ Alternate Scenarios
Test Data <ul style="list-style-type: none">▪ Data Requirements▪ Data Parameters
Services <ul style="list-style-type: none">▪ External Interfaces Requirements▪ API Definitions
Test Automation <ul style="list-style-type: none">▪ Framework linkage▪ Keyword linkage▪ Actions on the app

Roles and Benefits

Modeler (PO/BA/Tester)

- Draws and maintains business process flows
- Baselines understanding across the team
- Removes all ambiguities

Test Data Engineer

- Automatically generate synthetic data
- Find and reserve data automatically
- Match test data to each test case

Services Engineer

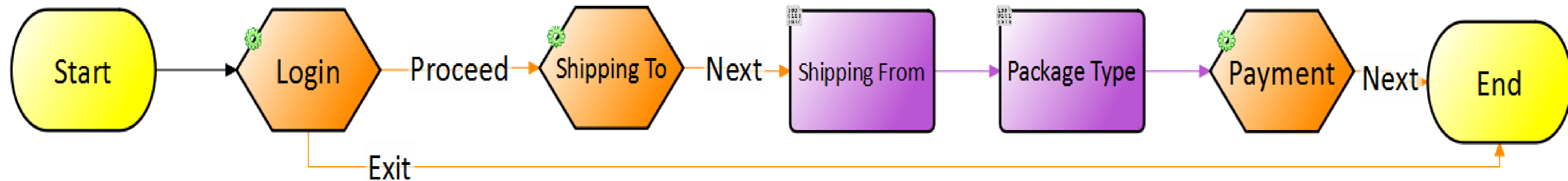
- No dependency on external interfaces
- Accelerate Code Development

Test Automation Engineer

- Automatic creation of scripts
- Automatic impact analysis

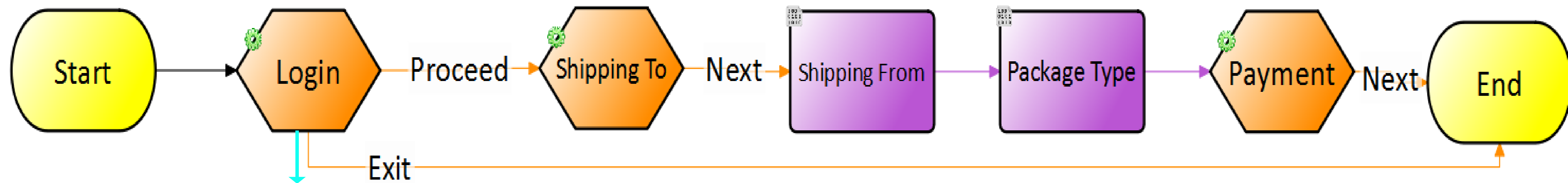
Business Analysts/Modeler – The “What”

Shipping Business Process

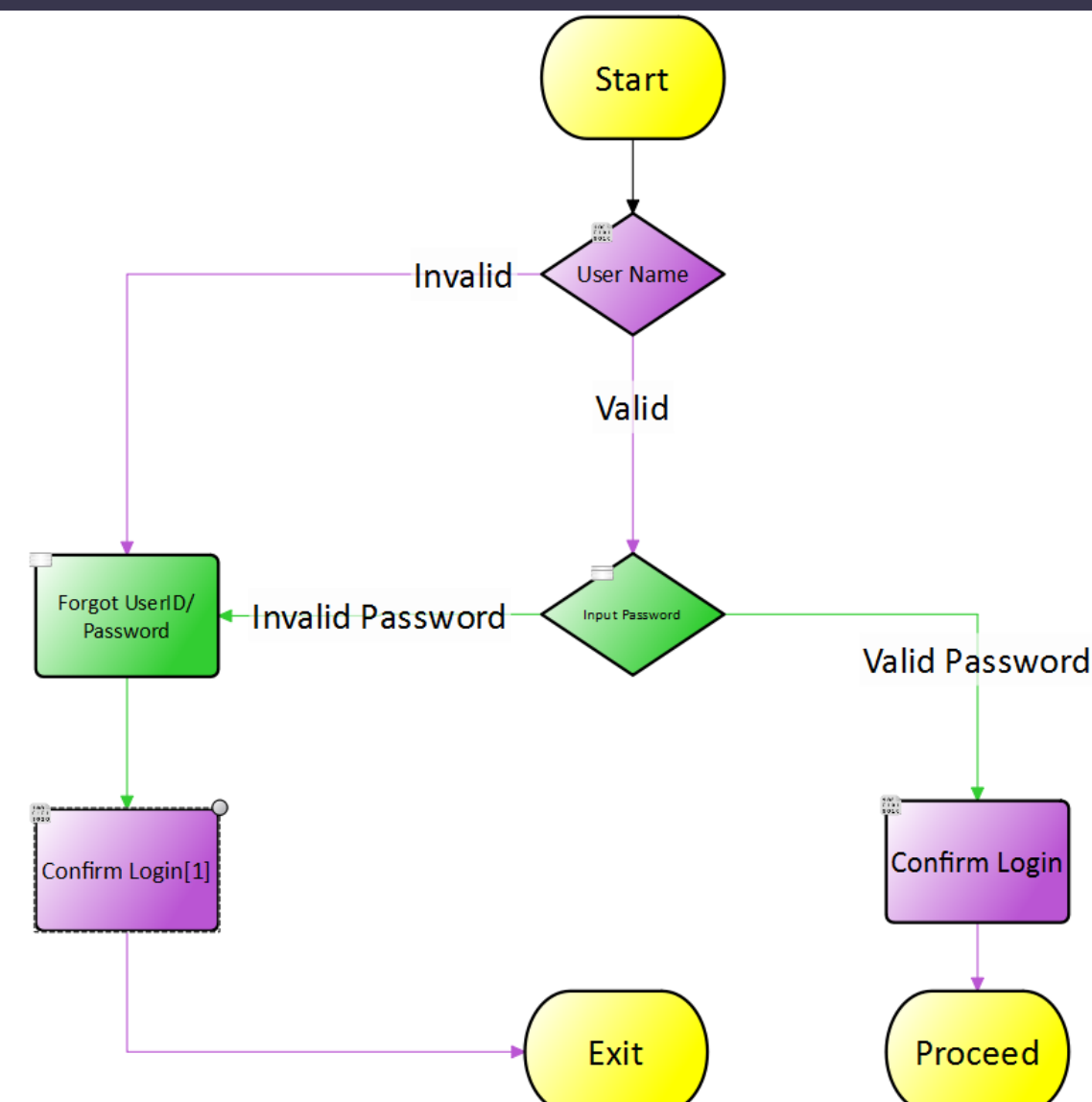


Quality Engineers / Testers – Foundation and Layers

Collaboratively enhancing the model



Model (Foundational Layer)



Test Data Layer

Services Layer

Test Automation Layer

```
1 Call Fn_EnterText(ObjIF.WebEdit("oUserId"),Parameter("~UserId~"))
2 Call Fn_EnterText(ObjIF.WebEdit("oPassword"),Parameter("~Password~"))
3 Call Fn_ClickButton(ObjIF.WebButton("bt_LogIn"))
4 '*****
5 If Browser("IF").Page("Fly with us").WebButton("Continue").Exist(3) Then
6     Call Fn_ClickButton(Browser("IF").Page("Fly with us").WebButton("Continue"))
7 End If
8
```

How do you build better apps faster?

CA BlazeMeter & Taurus



DEMOCRATIZING PERFORMANCE TESTING

SaaS and On-Premise performance testing platform
designed to help DevOps and Agile teams
begin web, mobile, micro-services, and API testing early in the SDLC
and re-use these tests all the way through to production.

Pure SaaS,
No Setup

100% Open-
Source
Compatibility
("Load and Go")

Public Cloud,
Virtual Private
Cloud and/or
Behind The
Firewall

Massively
Scalable
– 2 Million+
virtual users

Performance
Test
Orchestration
using Taurus

Comprehensive Reporting

Integration with APM, CI/CD. REST based API

Open-Source Technology – Enterprise Grade



- Open Source initiative sponsored by CA BlazeMeter
- Automation-friendly framework for Performance Testing
- Define scenarios in YAML or JSON
- Wide variety of test executors, including JMeter, Selenium, Gatling, and Locust
- Ability to generate load locally or in the cloud
- Consistent reporting through Taurus UI or BlazeMeter
- Easy integration with CI/CD in a SDLC
- <http://gettaurus.org>

```
scenarios:
  home-page:
    requests:
      - label: Home
        url: /
        method: GET
        think-time: 500ms
    assert:
      - contains:
        - Welcome
        subject: body
        regexp: false
```

Taurus Configuration Syntax

- Main section, declares tools to be executed during run, scenarios to use, etc

Execution

- Analysis and reporting settings, list of reporting modules to process results

Reporting

- Service modules configuration
- Shell execution
- pass fail criteria

Services

- Dictionary of scenario specifications with aliases, to be referred from executions

Scenarios

- List of classes to load and their respective settings

Modules

- Some top-level settings for the tool

Settings

- Advanced option, allows using resources other than local to have distributed high-load test, available if you have corresponding provider installed and configured

Provisioning

- This section allows to merge additional config files

Included-configs

```
---
execution:
- concurrency: 10
  hold-for: 5m
  ramp-up: 2m
  scenario: sample

scenarios:
  sample:
    headers:
      Connection: close
    requests:
      - http://localhost/

reporting:
- module: final_stats
- module: console

modules:
  jmeter:
    path: ./local/jmeter
    properties:
      log_level: DEBUG
  console:
    disable: false

settings:
  check-interval: 5s
  default-executor: jmeter

provisioning: local
```

Running Taurus

Command Line Tool

- Command-line tool is named **bzt** and invoked like
 - **bzt <options> [configs]**
 - **bzt taurus.yml -o modules.jmeter.path=alternate/jmeter/bin/jmeter -o provisioning=cloud**

Option	Expanded Option	Description
-h	--help	show help message and exit
-q	--quiet	only errors and warnings printed to console
-n	--no-system-configs	- skip /etc/bzt.d and ~/.bzt-rc
-v	--verbose	prints all logging messages to console (sometimes <i>a lot</i>)
-l LOG	--log=LOG	change log file location, by default is bzt.log in current directory
-o OPTION	--option=OPTION	override some of config settings from command line, may be used multiple times

How do you build better apps faster?

CA Agile Requirements Designer – Taurus Integration

CA Agile Requirements Designer – Taurus Integration

- CA Agile Requirements Designer creates:
 - Test Cases based on requirements
 - Test Data for the Test cases
 - Test Automation scripts for Test Case Execution
 - Configuration file to execute Test Automation scripts using Taurus and/or BlazeMeter
- Taurus is used to execute tests locally or in the Cloud
- Optionally, Taurus configuration file and Test Automation scripts can also be executed in CA BlazeMeter

Solution Benefits

- Improve application quality and testing efficiency using an optimal set of test cases for testing applications.
- Automatically generate fit for purpose Test Data using CA Test Data Manager to improve testing with a variety of test data
- Automatically generate Test automation scripts saving time and complexity
- Automate Test execution using Taurus and CA BlazeMeter to improve testing efficiency, reduce time to market and costs

Summary

- CA Agile Requirements Designer for requirements modeling, test cases and scripts
- CA BlazeMeter & Taurus for test execution on-premise and in the Cloud
- CA Agile Requirements Designer and Taurus integration helps improve test coverage and automate testing improving application quality and testing efficiency

How do you build better apps faster?

Demonstration

Thank You.

