

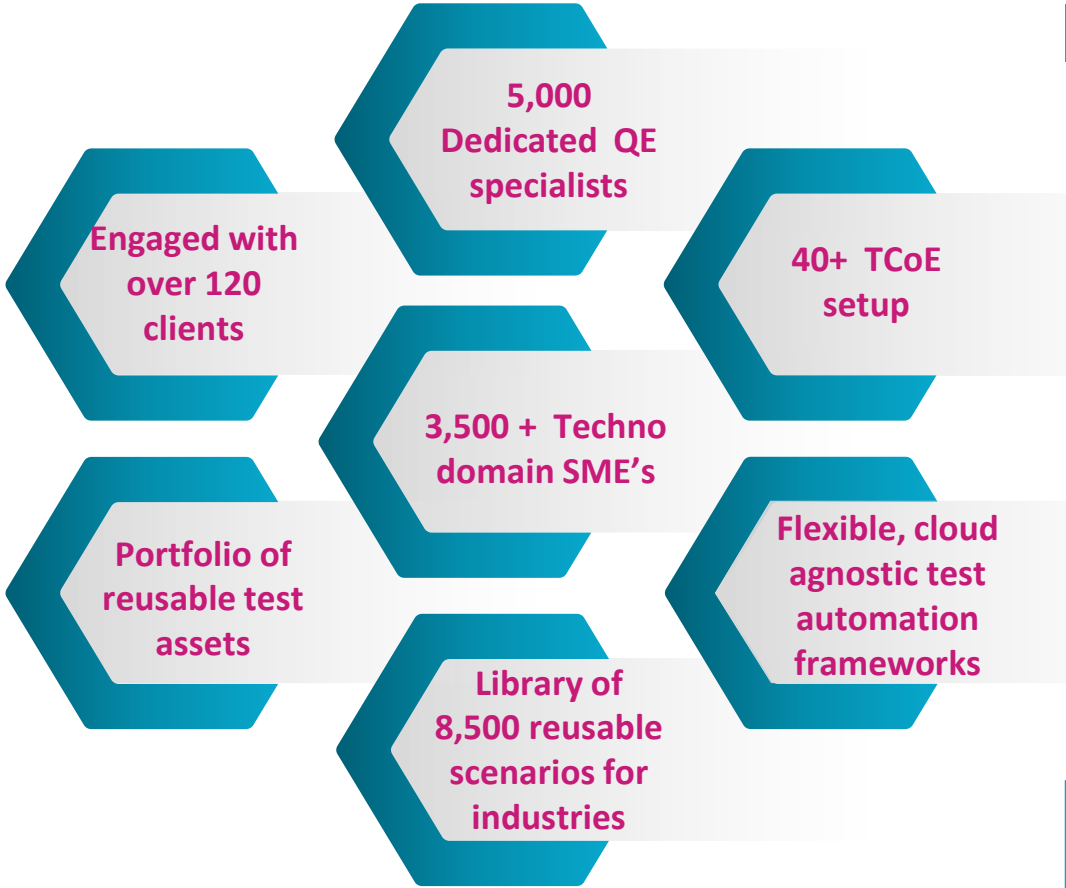


Mphasis Quality Engineering Services

Thursday, January 22, 2026



'Cognitive' Quality Engineering services



Quality Engineering Services

 Test Automation	 API Testing	 Native Mobile App Testing	 Performance Engineering	 Exploratory Testing
 End to End Testing	 Security Testing	 AI Augmented Testing	 Devices Testing	 Usability & Accessibility Testing
 Hardware Certification Testing	 Test Data Management	 AI Models	 Compliance Testing	 Cloud Migration Testing

Partners

Katalon

Sauce Labs

mabl

Synthesized

Tosca

ACCELQ

Perforce

Spec2TestAI

AI Powered Cognitive QE

Intelligent Test Design

Predictive Analytics

Self-Healing Automation

CQE

Automation Predictor

Schedule Predictor

Test Run Optimizer

Defect Predictor

Resource Allocator

AI-ML Frameworks

IP's & Innovations

Test Execution Automation

MAF

MSAFE

CQE

MSTF

MTAPE

UTAF

TAF

ATDD

Data Mgt

MDGEN

MDCERT

SynthStudio

Performance Engineering

PARGen

NFR Smart Analyzer



Our AI First QA to QE Transformation Approach focuses on 'Building Quality In' and Accelerating Delivery

AI First Quality Engineering

Infuse AI capability and use cases across the SDLC and all phases of Quality Engineering through AI platforms and tools.

Build Quality In, Shift from QA to QE

Build quality in rather than test it in. Integrate Quality Engineering practices across the entire SDLC from requirements to production.

Automate Everything

Automate all forms of test targeting 95% test automation. Utilize the pipeline to orchestrate this automation. Automate UI, Acceptance, API, Performance, Security and other test types.

Shift Left / Shift Right

Move testing as early in the process as possible.
Implement proactive production monitoring to ensure that escaped defects are detected as quickly as possible.
Code is delivered testable.

Frictionless E2E DevOps Pipeline

Implement an end-to-end DevOps pipeline to orchestrate all SDLC, quality, stage gates, and development related operational task.

Data Driven Continuous Improvement

Create multi-level dashboards to provide continuous visibility into the end-to-end DevOps & Quality process and progress against organizational KPIs relating to quality, agility, and the delivery of business value.

TDD, BDD, ATDD

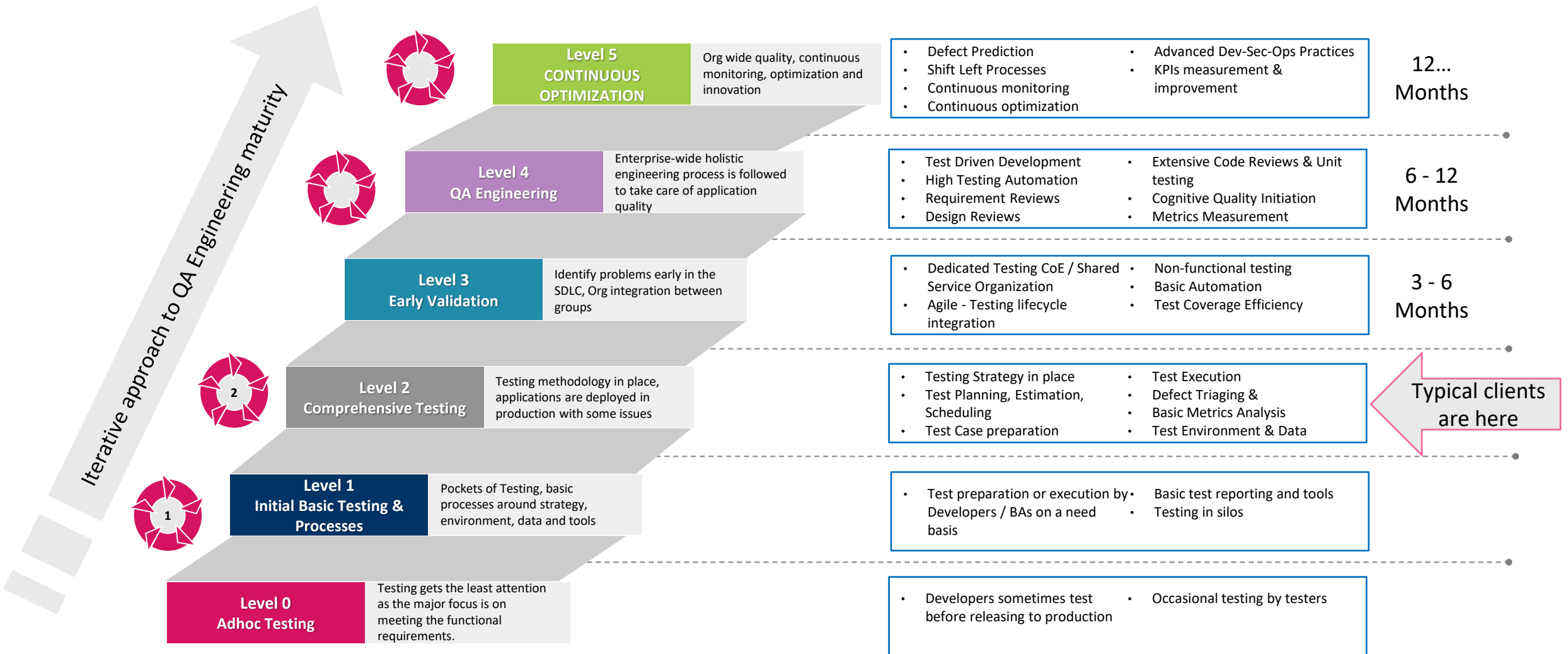
Adoption of solid test approaches and frameworks to support all phases of testing.

Test Environment & Test Data Mgt.

Streamlined Test Environment availability at par with production environment and Test Data availability, including Synthetic Test Data generation

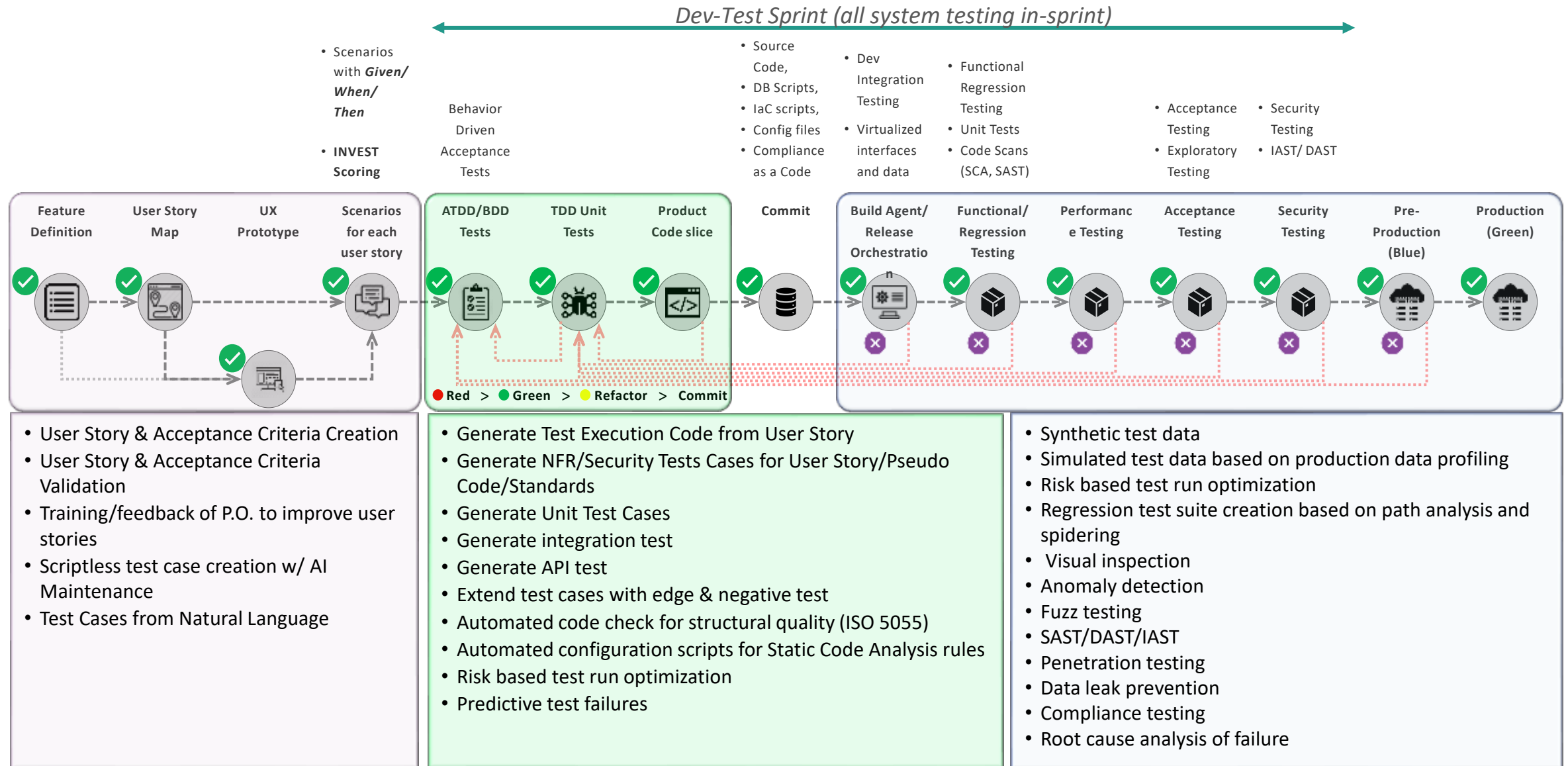


Typical stages towards attaining long term maturity for QE in enterprises





AI tools and interventions in Quality Engineering are evolving and enhancing with GenAI





AI Augmented QE Use Cases

1 Unit Testing

Generation of unit tests from ML models by analyzing code snippets



2 Defect Prediction

Identifies gaps in QA goals leveraging ML and NLP techniques using historical QA data



3 Requirements Management

Generates requirements automatically using natural language techniques, NLP, NLU or from Pseudocode



4 Customer Journey-Driven Testing

Automatic Generation of test steps that mimics production workflow using ML, Incorporates AI models to analyze production logs, Usage patterns, and personas



5 Scenario Driven API Testing

Generates Tests based on business logic using ML models to learn logical connection between test steps to create test scenarios



6 Requirements Driven Testing

Generates and analyze requirements, user stories and information captured by NLP and NLU



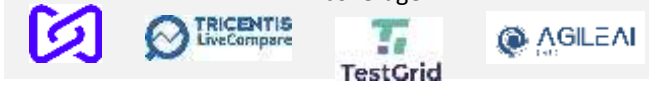
7 Test Case Selection

Leverages historical QA data to determine optimal test sets and risk indexes continuously refactored from execution cycles



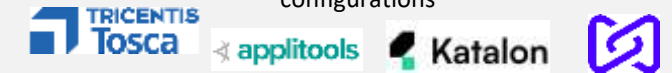
8 Test Set Optimization

Leverages CQE models to identify duplicates, redundant test cases, optimization of test execution sequence and gaps in test coverage



9 Self Healing

Leverages capabilities within AI enabled tools to identify changes to object properties. Endpoints, workflow and environment configurations



10 Auto Completion

Leverage AI / ML models and historical data to auto correct test scripts, input test data sets from test execution cycles



11 Performance Engineering

Create relevant and accurate performance models based on real user behavior by using ML to analyze data from real user monitoring and production logs.

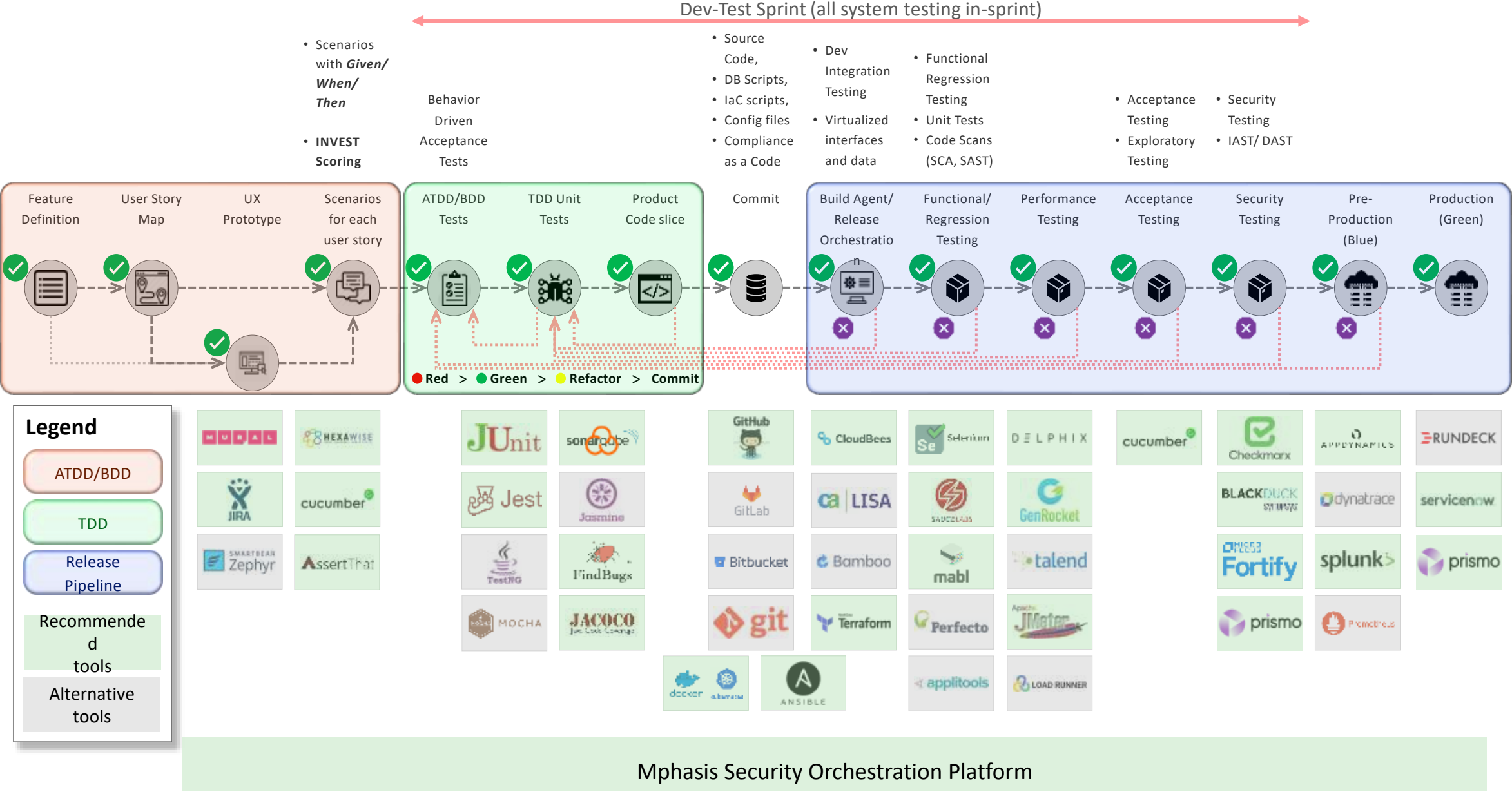


12 Test Data Generation

Generate large volume test data based on usage patterns and domain insights, cleanses and purges duplicate data sets.



Our differentiating Shift Left-QE Approach improves quality across the delivery value chain



Case Studies



AI Augmented QE Case Study for leading Global Hardware Manufacturer - HP

25% faster time to market

90% Increase in customer satisfaction

20% reduction in delivery cost

50% reduction in test scheduling & planning effort

Context/Scenario



- Product/Hardware Testing with agile release cycle of 2 weeks with CI/CT implemented
- Testing cycle has 2 SIT deployments per week & 1 deployment into UAT
- Post every deployment, team executes standard set of Regression Test Suite in SIT before deploying into UAT on Friday which is defined as Green TCs

Problem Statement



- Manually tracing through 7000 TCs and identifying optimal test suite for a release is time consuming and is prone to human error
- Manually prioritizing the TCs for a release which is a combination of manual and automation TCs
- Defect leakage
- Lack of coverage of edge cases

Solution



- Based on the historical data, the model designed which predicted the regression test cases that should be added to the test plan for a new requirement
- Based on historical patterns of defect occurrence, number of times test case was selected across past releases, and mandatory test cases, a priority order generated
- The model optimized the test suite based on the complexity and priority. The complexity is defined in the client's dataset against each test case

Benefits



- 75% reduction in SME/test lead bandwidth for the test suite generation
- 20% reduction in regression test effort depending on the feature
- 30% early defect detection

TCoE Transformation for A Global Insurer



Client Background

- A leading global General Insurance and Reinsurance company serving Australia, North America and Europe regions



Client Challenges & Asks

- Multiple legacy systems performing similar functions caused by acquisitions in various geographies leading to cost inefficiencies
- Migrating to a single global PAS and billing system – Majesco Operating Platform (MOP)
- Out of the box / custom features
- Needed exhaustive testing for multiple years
- Lack of QA Automation
- Key ask: Deliver optimum quality driven by efficient and effective automation testing approach



Engagement Highlights

3

Regions – NA, Europe and Australia

250+

Quality Engineers

100%

Regression Automated

>99%

Defect Removal Efficiency

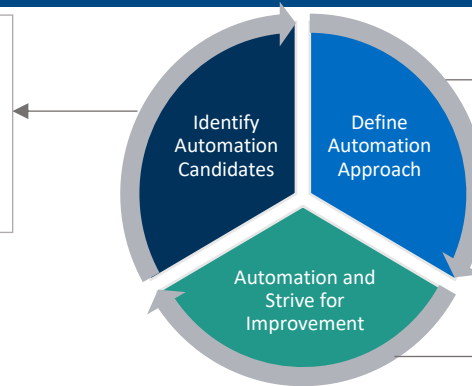
10+

Years engagement



Solution

- Common Product Flows
- Common Transactions
- Interfaces
- Rating Tests
- Regression and Smoke Tests



- Hybrid Tool Agnostic Framework - MAF
- Foundation - Reusable LOB / Functional Components
- Integrated with Test Management ALM tool

- Rating Test Accelerator
- Bulk Test Data Generation – MDGEN
- Functional Heat Maps
- Web Services test automation in Dev Env



Business Benefits

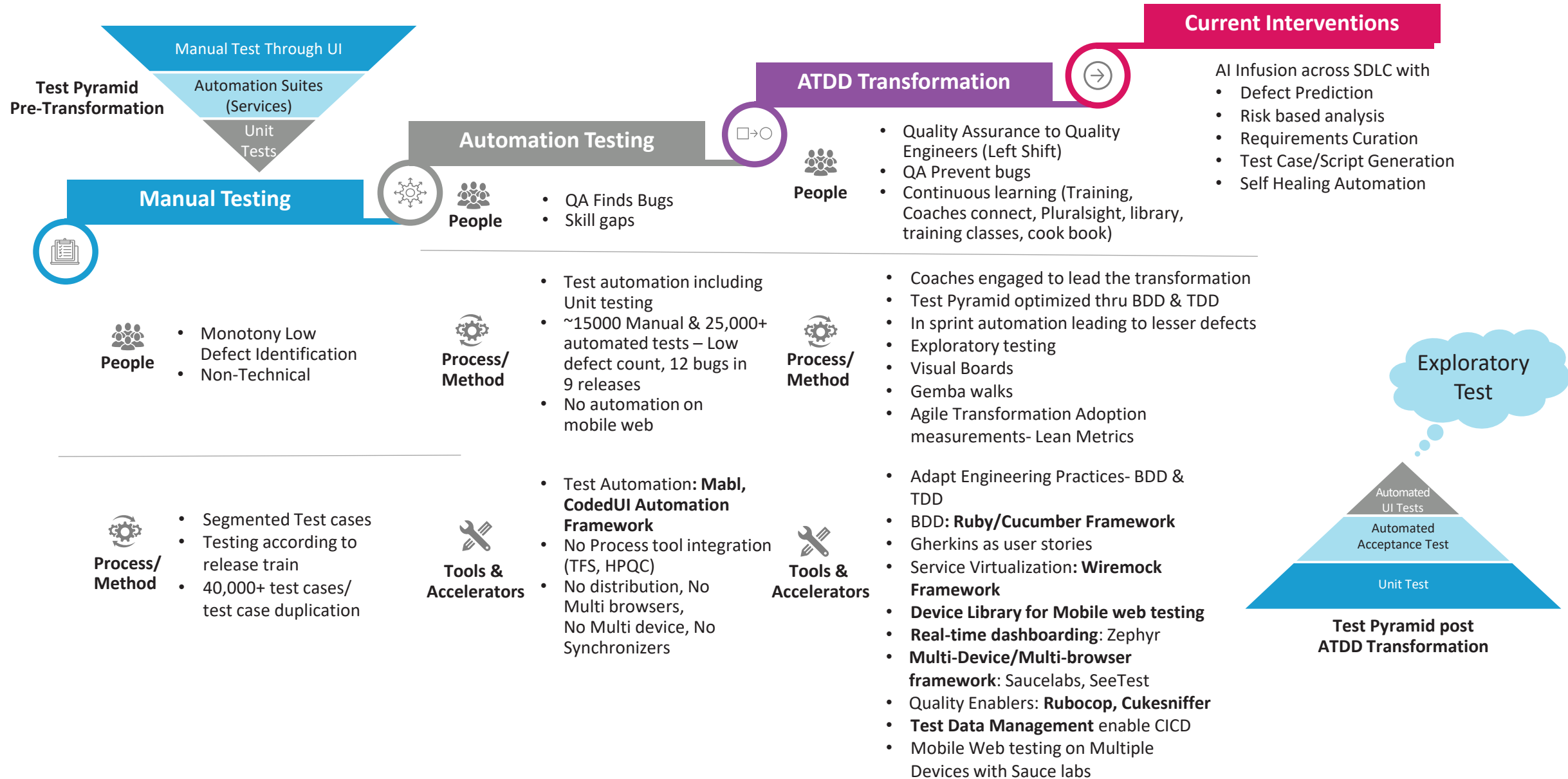
- Automation implemented using Tool agnostic framework brought down regression cost by 80%
- 8-12% Early Defect Detection; 5-8% Improved Time to Market
- Reduced test effort and improved test coverage



Learnings & Best Practices

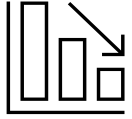
- Engaged Domain SME, Product SME, Client SME and Automation Test Automation Architect throughout the project
- Co-located team significantly improved productivity
- Functional Heat Maps and Reusable automation test packs together accelerated Regression test coverage
- Client University – Helped in rapidly bringing resources up to speed on domain, product and project

Mphasis Success @ Top American Brokerage and Banking Company



Mphasis Success @ Top American Brokerage and Banking Company

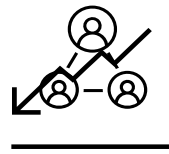
Reduction of 99% of defects over the product lifecycle



60-80% reduction in overall cycle time



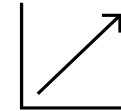
50% reduction in testing headcount



High quality with Zero or <1% defects in Production



Up to 80% saving in execution cycle and faster time to market



Examples of Business Benefits Realized in Projects

Time to Market

- **80% improvement in time to market with zero critical defects leaked to production.**
- **In-sprint Test : 1-2 Days at End of Dev sprint to 45 Minutes**
- **Regression test : Separate Sprint (3 – weeks*) to EoD (60 Minutes)**
- **UAT : 1 Day to 60 Minutes**

Quality

- **< 1% defects** in release hardening
- **< 5% in Sprint defects** due to well defined acceptance criteria

Overall Cost

- **95% automated test suite**
- **50% cost avoidance over 5 years**
- **60 % over all testing cycle reduction**

Success story: QA to QE Transformation for one of the top 4 US Bank

- Mphasis is involved in defining and executing the overall QE transformation journey, tools and frameworks
- QE adoption across 330 scrum teams
- Embedded 550 QE Engineers to enable the journey (QE engineers roll off to new scrum teams post maturity)
- As part of adoption, trained Scrum Teams to practice & achieve QE adoption (achieved 100% self sustaining maturity)

Transformation Levers



Transformation Impact

Time to Market

- Acceptance tests - **2 days to 45 mins**
- Regression tests - **3 weeks to 60 mins**

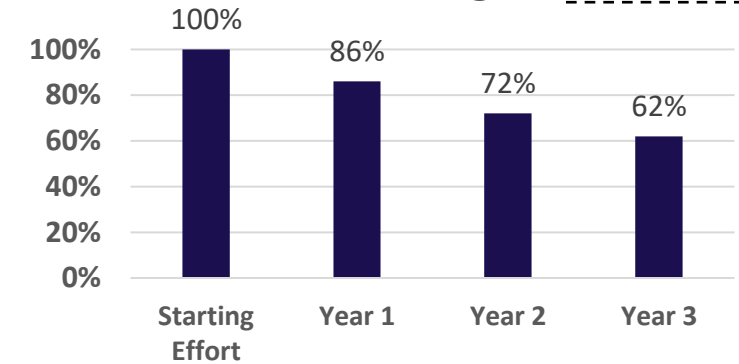
Quality

- **Less than 1% defect** in integration environment
- **Less than 5%** in-sprint defects because of well defined acceptance criteria

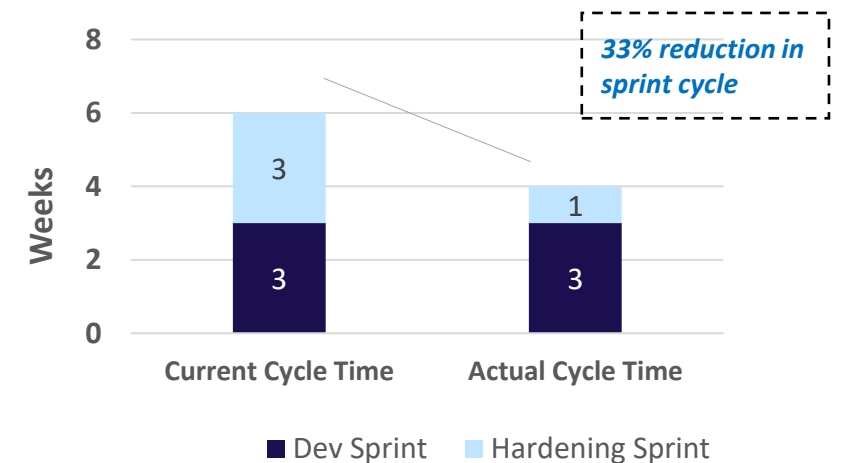
Talent

- **All resources are BDD trained engineers to handle stories end-to-end**

Effort Savings



Improvement in Cycle time



Thank You!

For more information, please contact:

About Mphasis

Mphasis (BSE: 526299; NSE: MPHASIS) applies next-generation technology to help enterprises transform businesses globally. Customer centricity is foundational to Mphasis and is reflected in the Mphasis' Front2Back™ Transformation approach. Front2Back™ uses the exponential power of cloud and cognitive to provide hyper-personalized (C=X2C2 TM=1) digital experience to clients and their end customers. Mphasis' Service Transformation approach helps 'shrink the core' through the application of digital technologies across legacy environments within an enterprise, enabling businesses to stay ahead in a changing world. Mphasis' core reference architectures and tools, speed and innovation with domain expertise and specialization are key to building strong relationships with marquee clients. Click [here](#) to know

Important Confidentiality Notice

This document is the property of, and is proprietary to Mphasis, and identified as "Confidential". Those parties to whom it is distributed shall exercise the same degree of custody and care afforded their own such information. It is not to be disclosed, in whole or in part to any third parties, without the express written authorization of Mphasis. It is not to be duplicated or used, in whole or in part, for any purpose other than the evaluation of, and response to, Mphasis' proposal or bid, or the performance and execution of a contract awarded to Mphasis. This document will be returned to Mphasis upon request.

Demo



AI powered end to end workflow

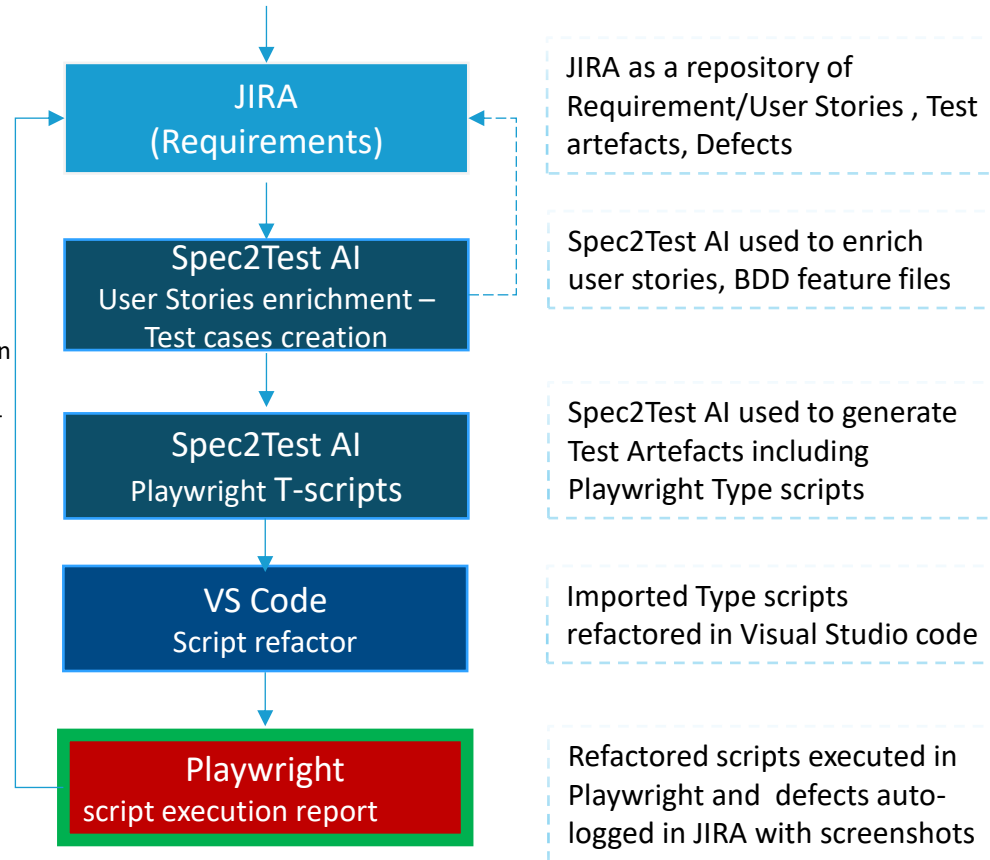
User Story used

Swag lab sandbox is leveraged-
for simple store to demonstrate
Login and adding product to
cart user stories from UI front

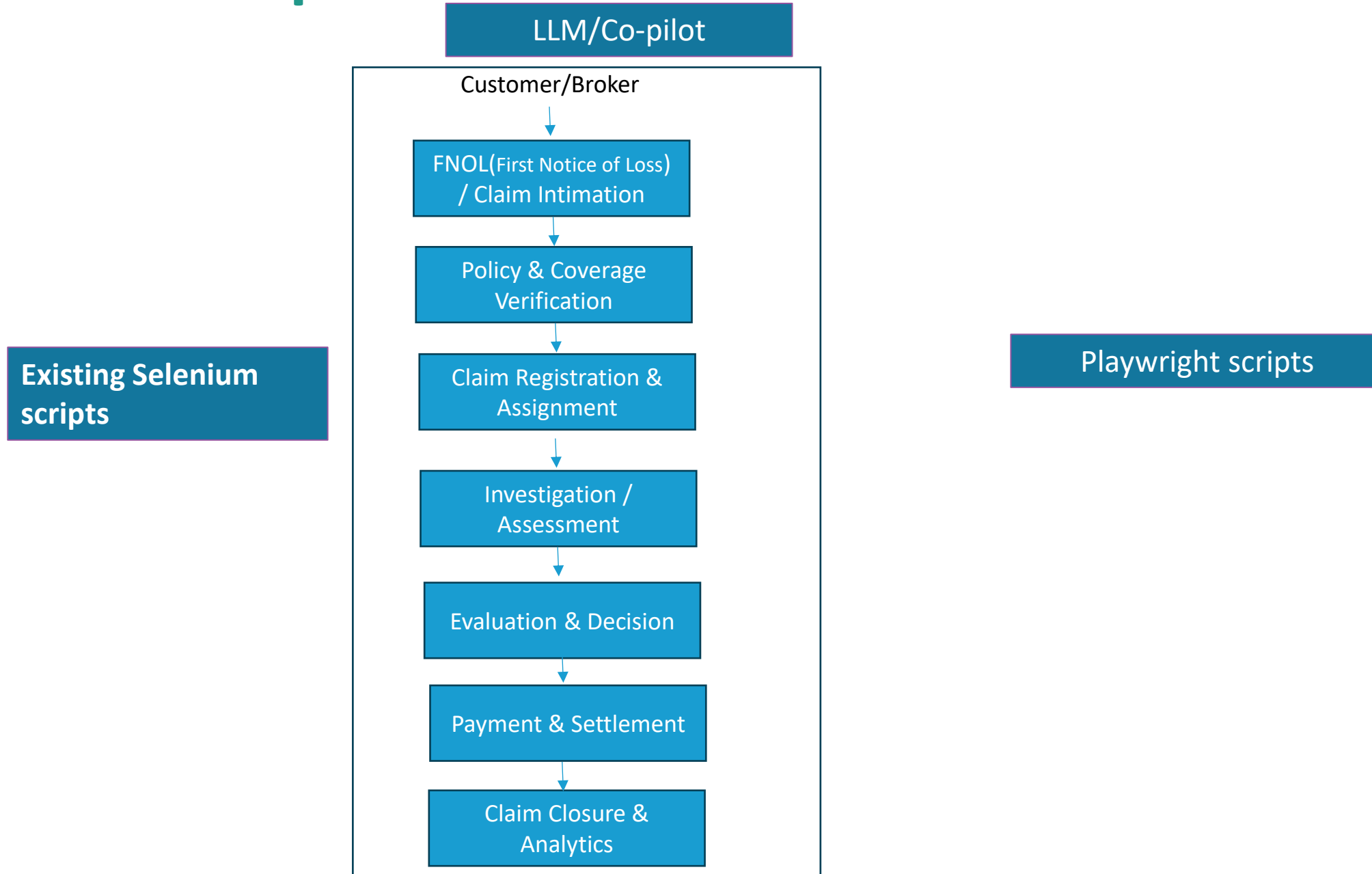
Auto-
logging
defects in
JIRA
(Defects-
logged
against
failed
test
scripts)

Para bank sandbox is used for
demonstrating pure API and
hybrid – UI/API user stories

To verify that the account
balance displayed in the Para
bank web UI - matches the
balance returned by the
backend API for the same
account.



POC – scope





Playwright Case Studies – Global Computer Manufacturer

About the System/Application: The embedded management processor on ProLiant and Apollo servers enables monitoring and troubleshooting, even if the OS is down, provided power and network are available.

Automation Scope: Playwright validated integration across modules like Onesight, Infosight, and GreenLake, supporting hardware control via the cloud. All cloud console use cases were tested with Playwright. Automated tests included factory defaults, alert mail, iLOREST, CLI set, AHS log analysis, QIT, OV QIT suite, and regression. A system was designed to consolidate reports from multiple formats (HTML, XLS, JSON) generated daily.

The proprietary Magik framework was developed for flexible reporting and monitoring

Volumetrics: 3000 test cases were automated. Git hub, commit triggers every night, weekly regression. Under release 30% would be executed.

Benefits:

Reduced execution time per configuration from 1 day to ~30 minutes – 94% reduction in execution time

Total execution time for 5 configurations reduced from 5 days to ~1 day – 80%

100% automated stress testing

Automated test case generation via CSV

Automated AHS log analysis using Aviator

Faster report processing and data access

Consistent output and elimination of manual errors