MODUS CREATE

STREAMLINING DEVOPS:
A STRATEGIC APPROACH TO TEST AUTOMATION

Joel Gerbino | October 27, 2023

JOEL GERBINO

Director

of Quality Engineering at
Modus Create
Joel.Gerbino@ModusCreate.com

https://www.linkedin.com/in/joel-gerbino-45495414/

25

years in Software Quality

Experience

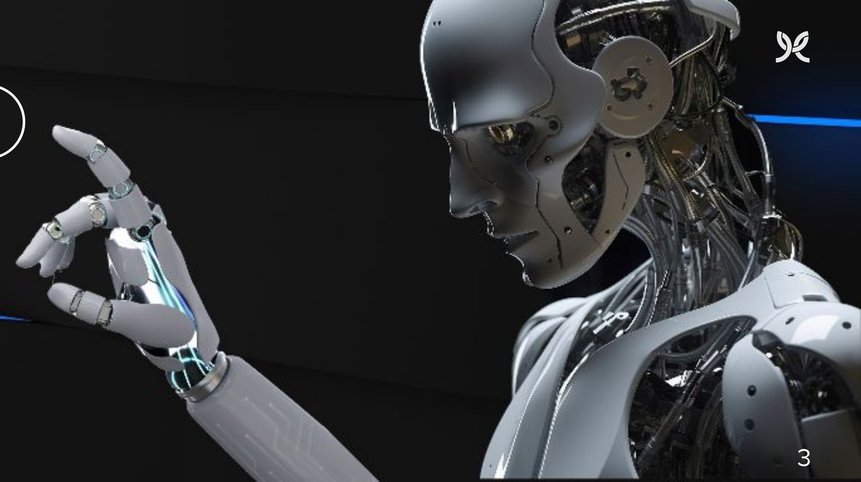
includes building and leading on-site and distributed teams, Agile transformations, designing and implementing automation solutions, and software quality process improvement

Lives

in Philadelphia Pennsylvania

The use of software to perform or support test activities (ISTQB)

WHAT IS TEST AUTOMATION?

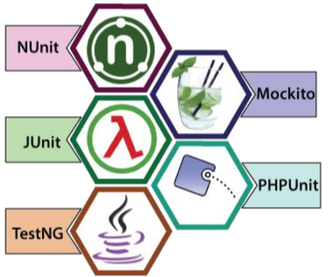


BENEFITS OF TEST AUTOMATION

- Less manual testing
- Increased test coverage
- Higher accuracy/repeatability
- Earlier discovery of defects
- Faster delivery of higher quality software

AUTOMATION TYPES AND TOOLS

Unit Testing



UI Testing

Appium
Selenium
Javascript Testing
Cypress
Playwright
Puppeteer

API Testing

TestGrid
Katalon Studio
SoapUI
TRICENTIS Tosca
REST-assured
Karate DSL
apigee
JMeter
Swagger
Assertible

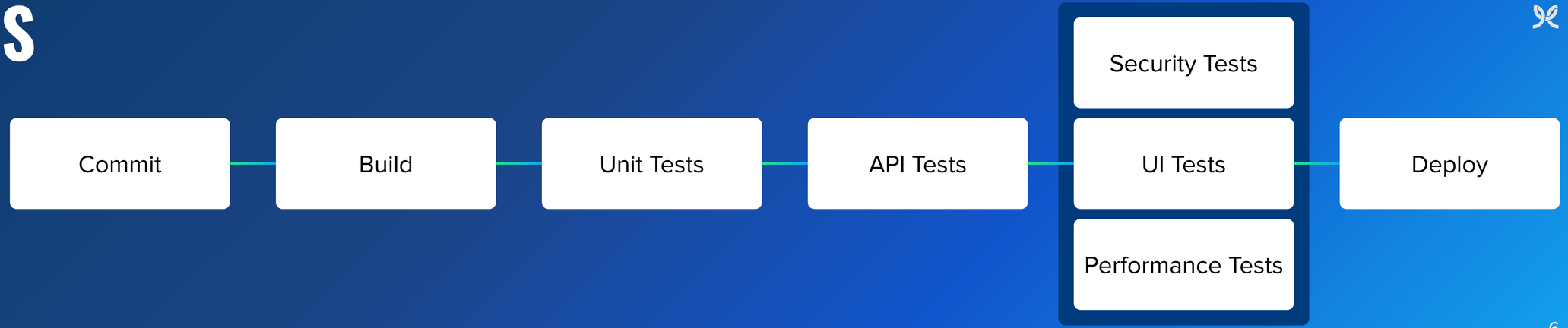
Security Testing

Burp Suite
VERACODE
Acunetix

Performance Testing

k6
JMeter
Gatling

CONTINUOUS TESTING



AUTOMATION STATISTICS

55%

of teams report achieving benefits from their test automation solution. (World Quality Report)

60%

of agile teams have test automation implemented. (World Quality Report, North America)

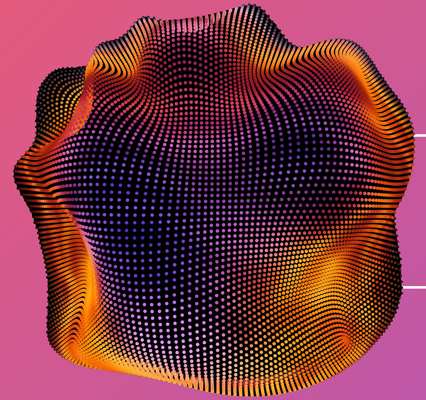
24%

Of test cases are executed using test automaton for many teams making Quality Assurance and Testing is a bottleneck to implementing DevOps and Continuous Delivery for many organizations (Continuous Testing Report)

BIGGEST CHALLENGES

- Companies lack skilled automation resources *(Perfecto, State of Test Automation 2023)*
- Teams prioritize selecting the test automation tools but don't define a proper test automation plan and strategy *(World Quality Report)*
- Test automation happening in silos, not integrated into the development process *(World Quality Report)*

IT'S COMPLEX!



Test automation spans well beyond setting up a framework and writing tests

It requires a well thought out strategy that intersects the SDLC, execution via DevOps pipelines, code management, collaboration, and many other factors

RUNNING BEFORE YOU CAN WALK

Selected a Framework

Automated Test

Executed Tests via Pipeline

~~Automation Strategy~~

NEGATIVE RESULTS

- Issues not caught by tests that should have caught them
- False failures triggered on-call alerts
- Stakeholders and team lost trust in automation

REBOUNDING FROM FAILURE

- Understand & address the issues
- Rebuild trust
- Build on success

WHAT WENT WRONG?

Framework Issues

+ Tests timing out and failing
+ Tests dependant on other tests
+ Tests colliding during execution
+ Test code commented out
+ Test results not meaningful

Code Management Issues

+ Automation code checked in breaking tests or whole framework

Pipeline Issues

+ Pipeline failing without tests running
+ Test tasks commented out in build definition
+ Notifications not configured correctly or turned off
+ Alerts triggered when tests didn't fail

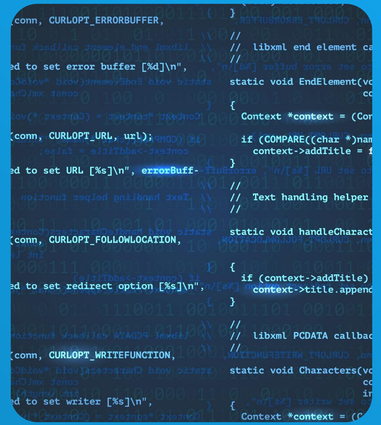
Working in Silos

+ Element locators changes breaking tests
+ Build agents and QA machines out of sync
+ No visibility into automation work

CHANGES INTRODUCED

New Framework

+ POM design pattern
+ Meaningful, actionable test results
+ Explicit waits
+ Headless test execution
+ Tests execution in all environments
+ Base test classes
+ Setup and teardown methods
+ Secure storage of credentials



CHANGES INTRODUCED

Pipeline changes

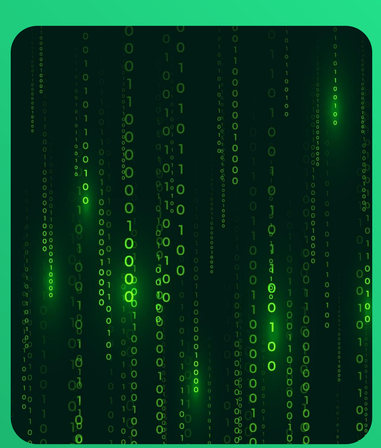
+ Permission policy changes
+ Reconfigured notifications
+ Pipeline troubleshooting guides
+ Pipeline build definition standards and templates
+ Azure dashboards for test results

CHANGES INTRODUCED

Coding standards & code management

+ Automated test and coding standards

+ Automation code management process



CHANGES INTRODUCED

Addressed the silos

+ Support process for DevOps issues
+ RCA/RCE processes for false failures and false positives
+ Product coding standards
+ Visibility/stories for all automation work
+ Overall communication greatly increases

AUTOMATING THE FIRST TEST

- Identified one critical test that would provide the most value
- Dedicated test account credentials that were stored securely
- Test executed in all environments from local machines or DevOps pipelines
- Screenshots captured as evidence of test coverage and upon failure

---

- Test results were meaningful and actionable
- Logging was implemented to provide detailed information throughout the execution of the test
- Dashboards were set up to display test results
- Slack webhooks were configured to trigger Slack notifications upon failure

PROVING VALUE AND EARNING TRUST

Monday Stats:

720	714	6
Total runs	Total passed	Total failed

Tuesday Stats:

720	720	0
Total runs	Total passed	Total failed

Cumulative Stats:

5,040

Total consecutive reliable executions

*All 6 failures were legitimate and were due to Monday's major incident

ROOT CAUSE ANALYSIS & ERADICATION

A root cause analysis was performed on all failures

Root Cause of Failure	Root Cause Elimination
Document service was down	NA - Failure was legitimate
Build canceled and triggered alert	Notification updated to trigger on failed test not failed build
Document upload failed	NA - Failure was legitimate

ONE VALUABLE TEST

- Caught several issues
- Notified on-call developers immediately
- Greatly reduced negative impact to customers

BUILDING ON THE SUCCESS

The process followed for the initial test paved the way for continued success

Walk

+ Performed Analysis
+ Developed Strategy
+ Developed Processes & Standards
+ Designed and Implemented Framework
+ Automated first critical test

Jog

+ Added a dedicated automation resource
+ Achieved 100% automation coverage of critical features

Run

+ Regression suite automated
+ Automation shifted left (new dev)

Sprint

+ Implemented an automation solution for compatibility testing
+ Began rolling automation processes out to other project teams

KEYS TO SUCCESS

Framework & tool selection

+ Development framework and techstack
+ DevOps tool integration
+ Test management integration
+ Support/community

KEYS TO SUCCESS

Framework creation

+ Structure i.e. POM design pattern
+ Coding standards i.e. naming conventions
+ Waits (explicit vs implicit)
+ Setup and Teardown methods
+ Base classes
+ Test data management
+ Execution in multiple environments
+ Logging
+ Assertions
+ Results / Evidence i.e. meaningful and actionable
+ Security
+ Reporting
+ Parallel execution
+ Maintenance i.e. package updates

KEYS TO SUCCESS

Test creation

+ What to automate and what not to automate
+ Test level i.e. Test Automation Pyramid
+ Test size i.e. tests are atomic
+ Test dependencies i.e. tests are autonomous

KEYS TO SUCCESS

Test execution & results

+ Execution schedule
+ CI triggers
+ Investigation process
+ Visibility of test results
+ Notification capabilities i.e. Teams, Slack, on-call alerting

KEYS TO SUCCESS

DevOps / Pipeline

+ Security i.e. secure storage and access
+ Pipeline timeouts
+ Test result storage & retention windows
+ Dashboards
+ Permissions i.e. test data cleanup, external environments
+ Logging
+ Resources i.e. adequate build agents
+ Parallel execution
+ Execution in multiple environment
+ Maintenance schedule & impact i.e. agent browser updates

KEYS TO SUCCESS

Code management

+ Branching i.e. main branch is restricted
+ PRs & Code reviews
+ Testing i.e. CI triggers for testing the framework

KEYS TO SUCCESS

Support & collaboration

+ DevOps team support i.e. support contact and process
+ Security team support i.e. password management
+ Development team i.e. Product coding standards, test data management

KEYS TO SUCCESS

SDLC

+ Automation for new development is part of the SDLC
+ Automation work is visible and included in project planning
+ Automation considerations are discussed during design and review sessions


Q
A

MODUS CREATE

THANK YOU!

Let’s make something great together.