DataArt

DataArt

Software Testing – Part 1

Ernesto Moyano
Senior QA & Developer

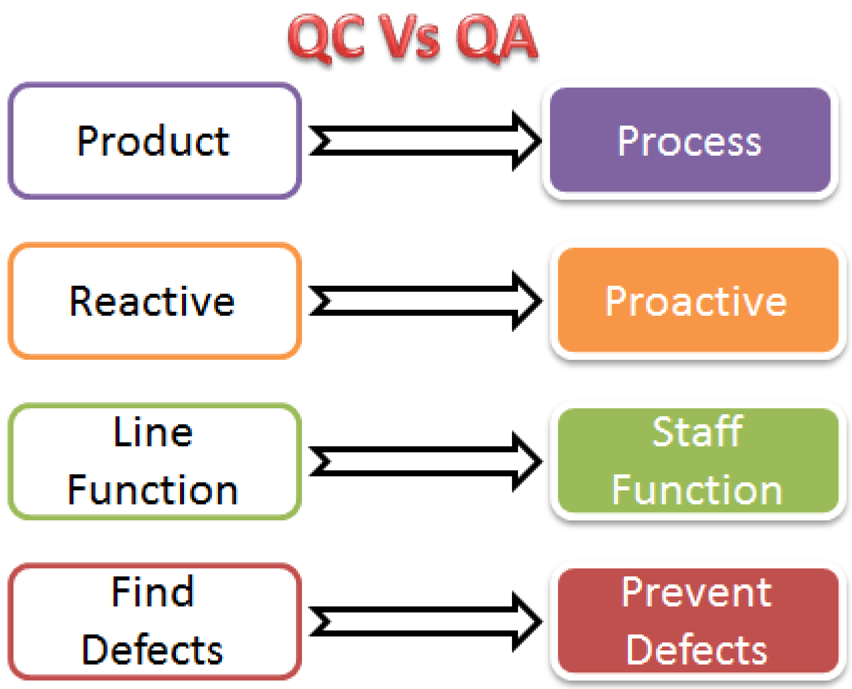
Pablo Gutierrez
Senior Developer

Agenda

1. General Introduction
2. Approaches & Strategies
3. Levels
4. Developers: Our scope
5. Tools

Software Testing – General Introduction

- Experimentation from *Scientific Method*:
  - Observation, Hypothesis, Experimentation.
- Early days: very manual. *Waterfall process*:
  - *Requirement, Analysis, Design, Develop, Test (*), Deploy.*
- Enterprises understood that "*Quality is money*" *(i.e. Slots Machines)*:
  - Not only product quality, but also regulations.
- As software process have *changed/evolved*, testing was getting ahead in the process timeline:
  - From the very end to "before that", ideally at the beginning.
- From *Quality Control* to *Quality Assurance* (taken from here)



Software Testing – Approaches & Strategies

DataArt

- Manual vs Automated Testing:
  - Automated test is the way to go in most cases
  - There might be certain situations (devices/hardware), we must go manual
- Approaches:

Static	Code analysis
Dynamic	Exploratory, running and expecting result
Passive	Waiting for things to happen, like in Data Engineering

- “Box” approach: White-box, Black-box, Grey-box.
- TDD, BDD _(**next webinar)._

Software Testing – Levels

- *Unit testing*: Testing in complete isolation.
- *Integration Testing*: Testing more than one (method/function inside the same artifact **or not**):
    - Component Testing.
    - System Integration Testing.
    - End-to-End Testing.
- *Acceptance Testing*: Functional:
    - User → Regression testing.
    - Business → Regression testing.
    - Usability.
    - Alpha / Beta Testing.
    - Smoke Test / Regression.
    - AB Testing: Pilot production test.

Software Testing – Levels *(continued)*

- Acceptance Testing: Non-functional Operational & Security
    - High Availability.
    - Performance.
    - Resilience.
    - Scalability.
    - Documentation.
    - Certification: i.e. PCI (Payment Card Industry Data Security Standard).
    - Vulnerability Scan.
    - Penetration Testing.
    - Security Review.
    - Security Procedures.

Software Testing – Comparison

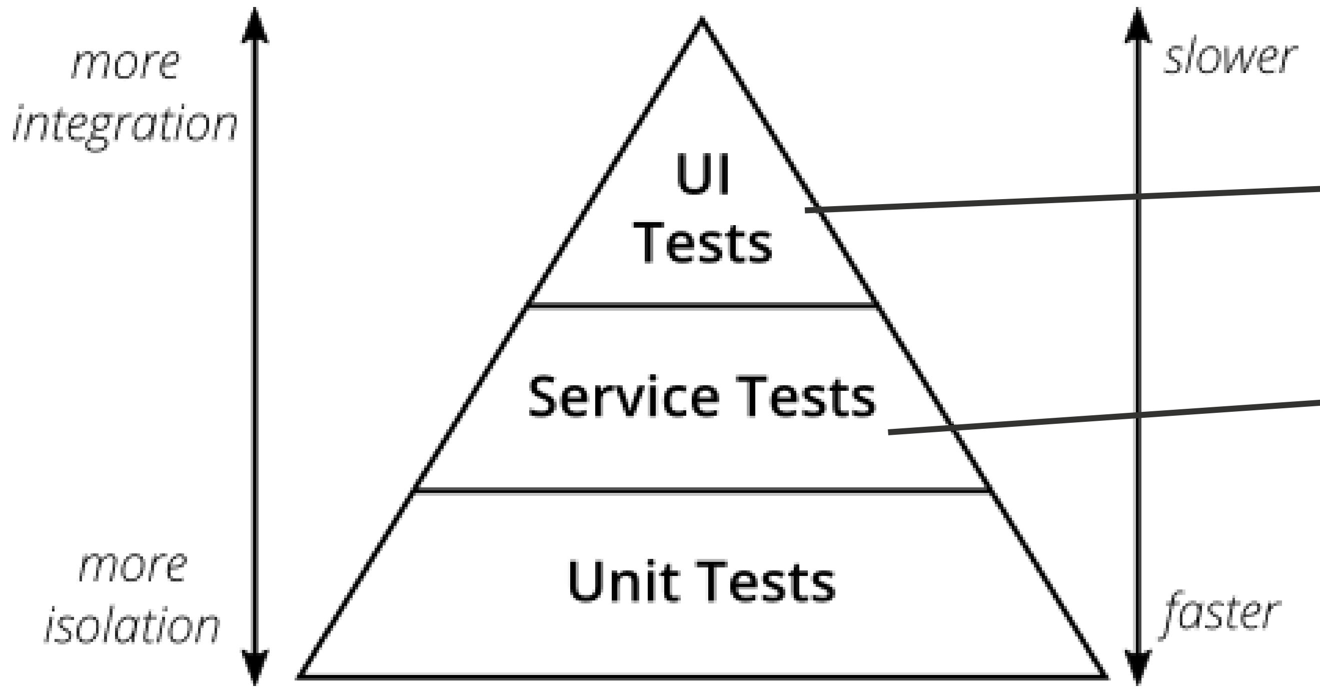
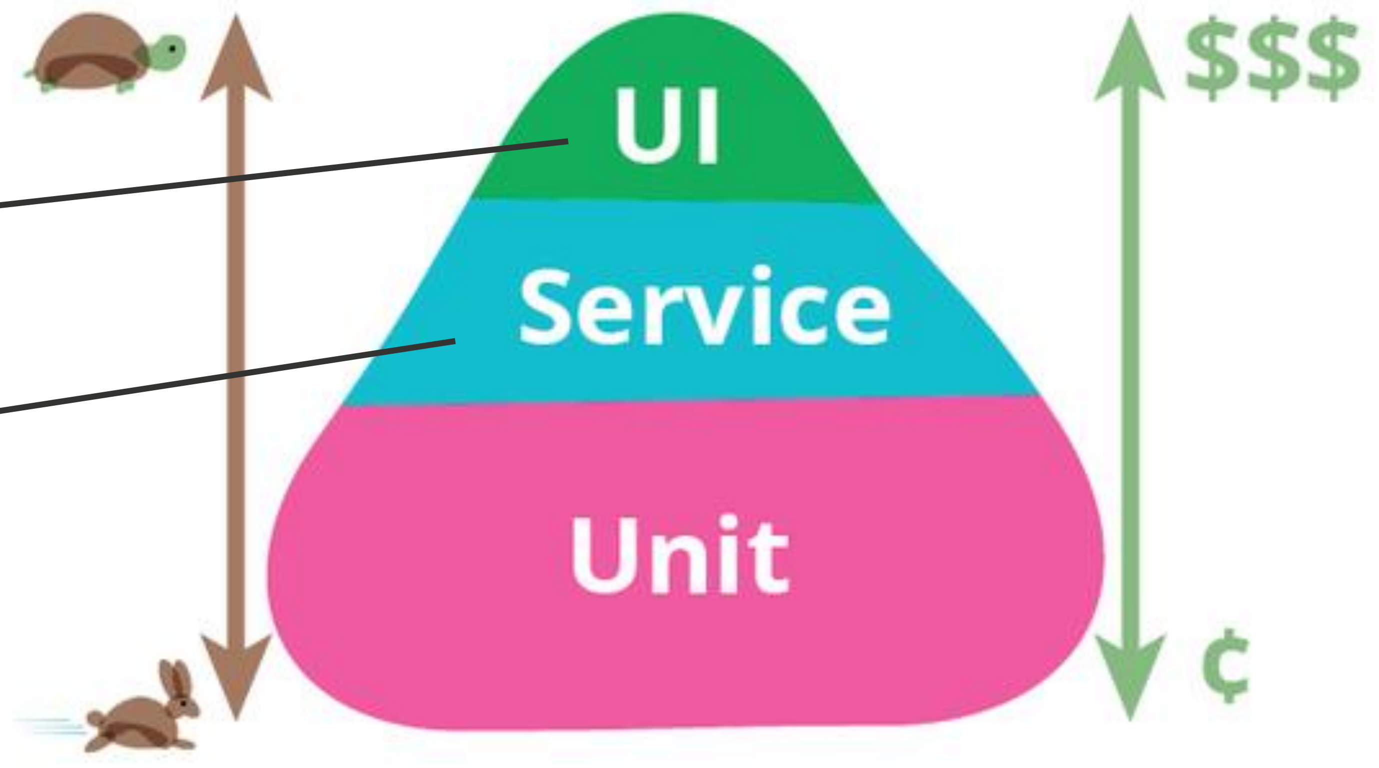


Figure 2: The Test Pyramid

Wrong: Should say

Acceptance Test

Integration Test

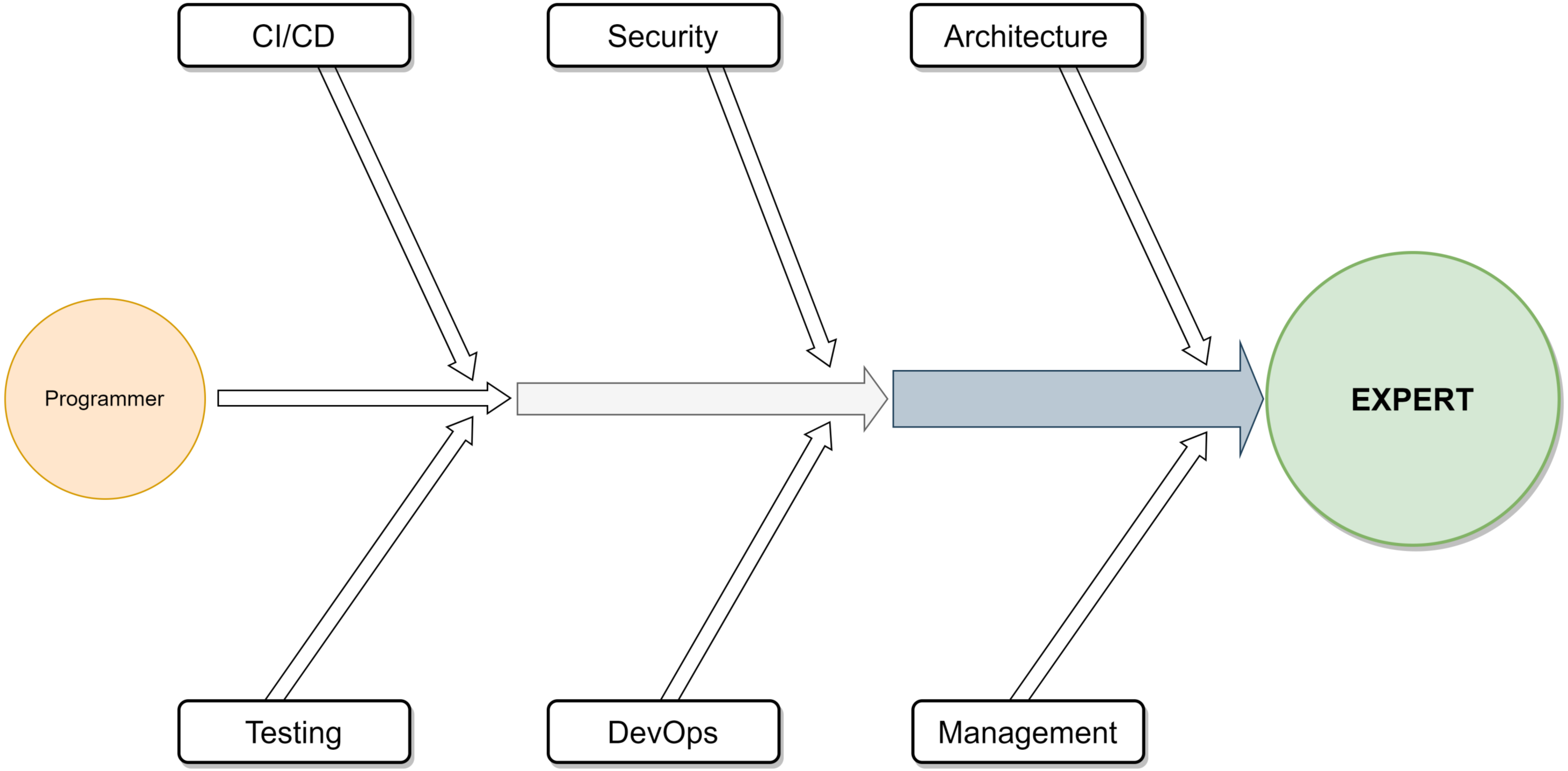


Taken from martinfowler.com

Software Testing – Why Devs?

DataArt

Why should we “*programmers*” learn about Testing?



Functional Testing tools

SMARTBEAR ReadyAPI

Katalon

TestGrid

apigee

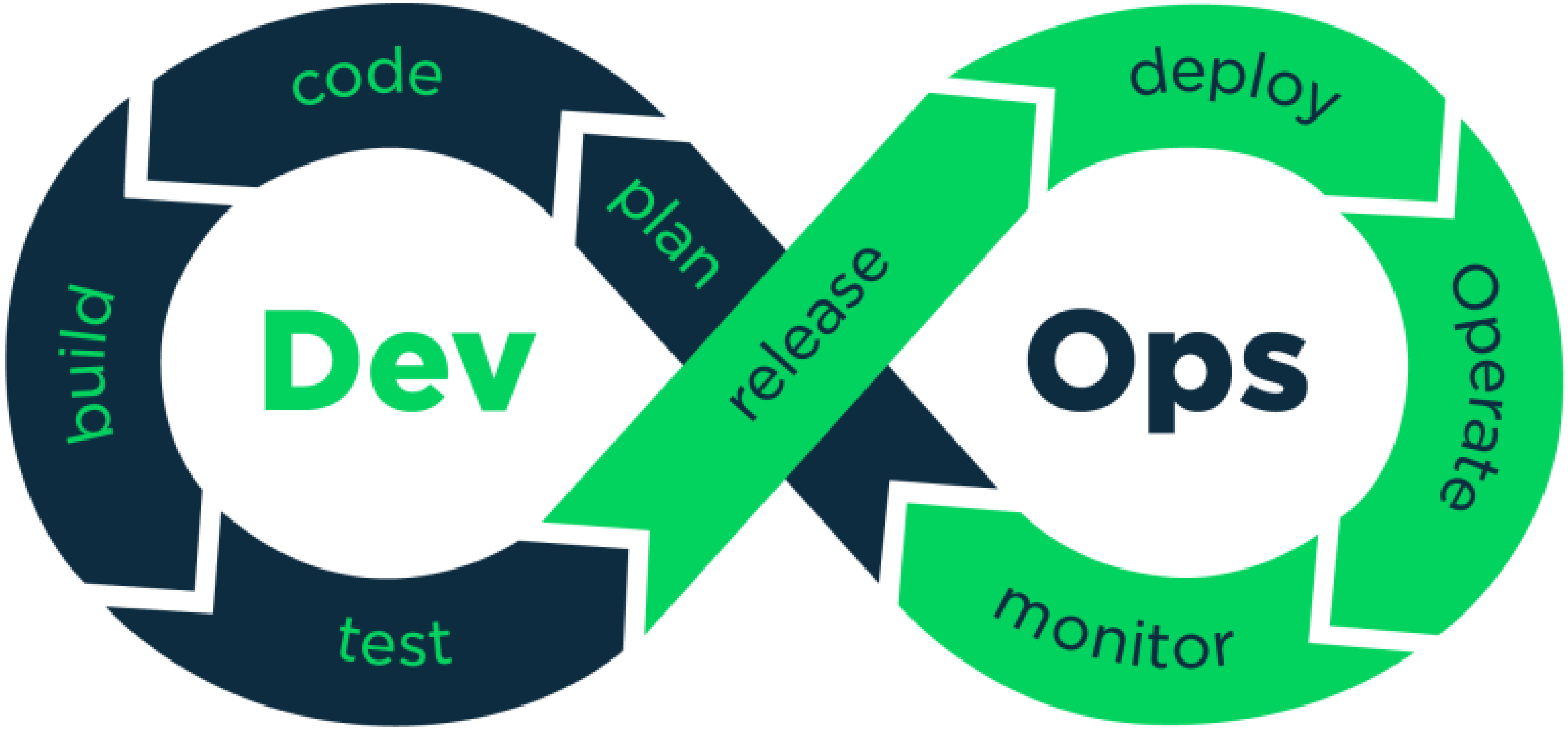
REST ASSURED

Swagger™

SoapUI

POSTMAN

Integration with CI/CD



Performance Testing tools

LOCUST

APACHE

JMeter™

SMARTBEAR

LoadNinja

Security Testing

BeEF

sqlmap

Google Nogotofail

sonarqube

ZAP

VERACODE

Foca

OPENSOURCE

Testing in Java – Next Webinar *(WIP)*

Testing in Java:

- Junit.
- Mockito (jMock, EasyMock).
- PowerMock.
- SpringTest.
- MockMvc.
- JaCoCo.

THANK

YOU !