

TESTING AND QUALITY RADAR - 2022

Powered by
thoughtworks

Techniques

ADOPT

1. Integration Testing (isolated)

Testing Strategy where suites are ran against fully deployed services (environment is not necessarily complete)

Note: use sparingly, and have a solid foundation of faster, more direct tests.

2. BDD

See [BDD Testing](#)

[<https://takeofftech.atlassian.net/wiki/spaces/~538152438/pages/3428876289/DRAFT+-+Quality+and+Testing+Guidelines#Behavioral-Driven-Design-\(BDD\)>](https://takeofftech.atlassian.net/wiki/spaces/~538152438/pages/3428876289/DRAFT+-+Quality+and+Testing+Guidelines#Behavioral-Driven-Design-(BDD))

5. End 2 End Testing (integrated)

Testing Strategy where suites are ran against fully deployed environments

A reasonable way to build confidence, but must be limited to key flows and well maintained

TRIAL

4. TDD

A development model where tests are written before implementation.

8. Contract Testing

Contract testing is a methodology for ensuring that two separate systems (such as two microservices) are compatible and are able to communicate with one other. It captures the interactions that are exchanged between each service, storing them in a contract, which can then be used to verify that both parties adhere to it. Contract testing goes beyond schema testing, requiring both parties to come to a consensus on the allowed set of interactions and allowing for evolution over time.

ASSESS

3. Log Replay

Supported by tools like [GoReplay <https://github.com/buger/goreplay/>](https://github.com/buger/goreplay/) log replay involves forms of capturing and replaying live HTTP traffic (or from logs) into a test environment in order to continuously test your system with real data.

HOLD

6. Mocks

Using stand-ins or "fakes" to simplify setting up automated tests

7. Feature Flags

Feature flagging (also commonly known as feature toggling) is a software engineering technique that turns select functionality on and off during runtime, without deploying new code.

Platforms

ADOPT

9. GitHub Actions

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform integrated with Github

12. Serverless (Cloud run and functions)

ASSESS

11. Cloud Build

Continuous integration and deployment process execution from GCP.

HOLD

10. Azure Pipeline

Continuous integration and deployment process execution from Microsoft's cloud.

Tools & applications

ADOPT

13. Testrail

TestRail is hosted test case and test management software with reporting and an API

15. Release Qualification Tools

Release Qualification Tools <<https://github.com/TakeoffTech/release-qualification-tools>> is a collections of python scripts that offer a form of guided release qualification. These are intended for semi-automated verification of deployments.

16. Integration Tests

Integration Tests Java Framwork and Tests Suites
<<https://github.com/TakeoffTech/integration-tests>> is a complex collection of End to End tests and supporting libraries written in Java. These tests cover a wide swath of legacy services. They are listed in the "tools and tests" quadrant becuae this entry is focused on the test/execution side and nothing more. **Only intended for legacy services. "HOLD" for new services**

TRIAL

17. PactFlow

PactFlow <<https://pactflow.io/>> improved version of the open source Pact Broker.

18. Pact Broker

Pact Broker <https://github.com/pact-foundation/pact_broker> is an application for sharing of consumer driven contracts and verification results. It is optimised for use with "pacts" (contracts created by the Pact framework), but can be used for any type of contract that can be serialized to JSON.

ASSESS

14. GoReplay

GoReplay <<https://github.com/buger/goreplay/>> captures and replays live (production) HTTP traffic into a test environments.

Languages & frameworks

ADOPT

19. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

20. GO

Go is a statically typed, compiled programming language designed at Google by Robert Griesemer, Rob Pike, and Ken Thompson.

23. Pytest

Popular Python testing framework that uses fixtures

TRIAL

21. Karate

>Karate is the only open-source tool to combine API test-automation, mocks, performance-testing and even UI automation into a single, unified framework. The BDD syntax popularized by Cucumber is language-neutral, and easy for even non-programmers. Unfortunately, the step implementation language is best done in Java or JavaScript.

25. GoDog

[GoDog <https://github.com/cucumber/godog>](https://github.com/cucumber/godog) is the official Cucumber BDD framework for Golang

26. Pytest-BDD

[Pytest-bdd <https://github.com/pytest-dev/pytest-bdd>](https://github.com/pytest-dev/pytest-bdd) is a BDD library for the py.test runner

HOLD

22. Java

Java is a high-level, class-based, object-oriented programming language

24. Serenity BDD

Another BDD Framework, Java based, not as compelling as Karate.