

Quality Issues Impact Velocity of Continuous Integration/Continuous Deployment Pipelines

Issue

As organizations advance in their DevOps programs to migrate toward full continuous integration (**CI**) and continuous deployment (**CD**), test automation is a must for ensuring productivity and product quality. Many organizations, however, have a high percentage of manual testing and a track record of failed automation attempts due to these underlying issues:

- Lack of a framework for ensuring patterns and reuse for test automation
- High licensing costs for commercial testing tools
- Use of tools or scripts that developers don't use (e.g., heavyweight, proprietary, not extensible)
- Lack of proper budgets and time for developing integrations for test management and defect tracking tools

Additionally, the following broader issues may need to be addressed as part of test automation framework implementation, as part of DevOps initiatives:

- Lack of continuous build (**CB**) and **CD** of the environment and application(s) under test
- Lack of a test data strategy
- Lack of a test management process
- Lack of quality standards

Lacking a strategy and framework for implementing test automation within your DevOps program will lead to delays or failed **CI** and **CD**, which puts your systems at risk, drives up costs, and/or causes you to miss revenue opportunities.

Elyxor Solution

The good news: Our team developed the Elyxor Test Automation Platform (**ETAP**) as an accelerator to improve quality and accelerate development and implementation of test automation for your projects. **ETAP** comprises the test services and utilities we have found to be reliable and in widespread use. The **ETAP** framework is versioned and allows for custom adapters and features as required, if they do not already exist.

Key **ETAP** features include:

- Support for test suites written in Java, Python, and .NET
- Supports for popular test engines, such as TestNG, Cucumber and pytest
- Integration with test management and defect management tools – Services use API calls to determine tests to run and for publishing execution results
- DevOps adoption by integrating with **CI** tools and versioning systems
- Elyxor Vorteks integration – Elyxor Vorteks platform provides a central portal and data store for all pipeline metrics used as the information source for visualizing the quality, velocity, and status of code delivery; tracking audit activity; and evaluating business rules for pipeline quality gates. (See Vorteks Test Automation Dashboard.)
- Pre-built libraries for test suite configuration and test data

As depicted in Figure 1, **ETAP** is a component within a test automation system, functioning as a bridge from application test suites to test suite configuration and data, testing tools, and test reporting (such as Vorteks).

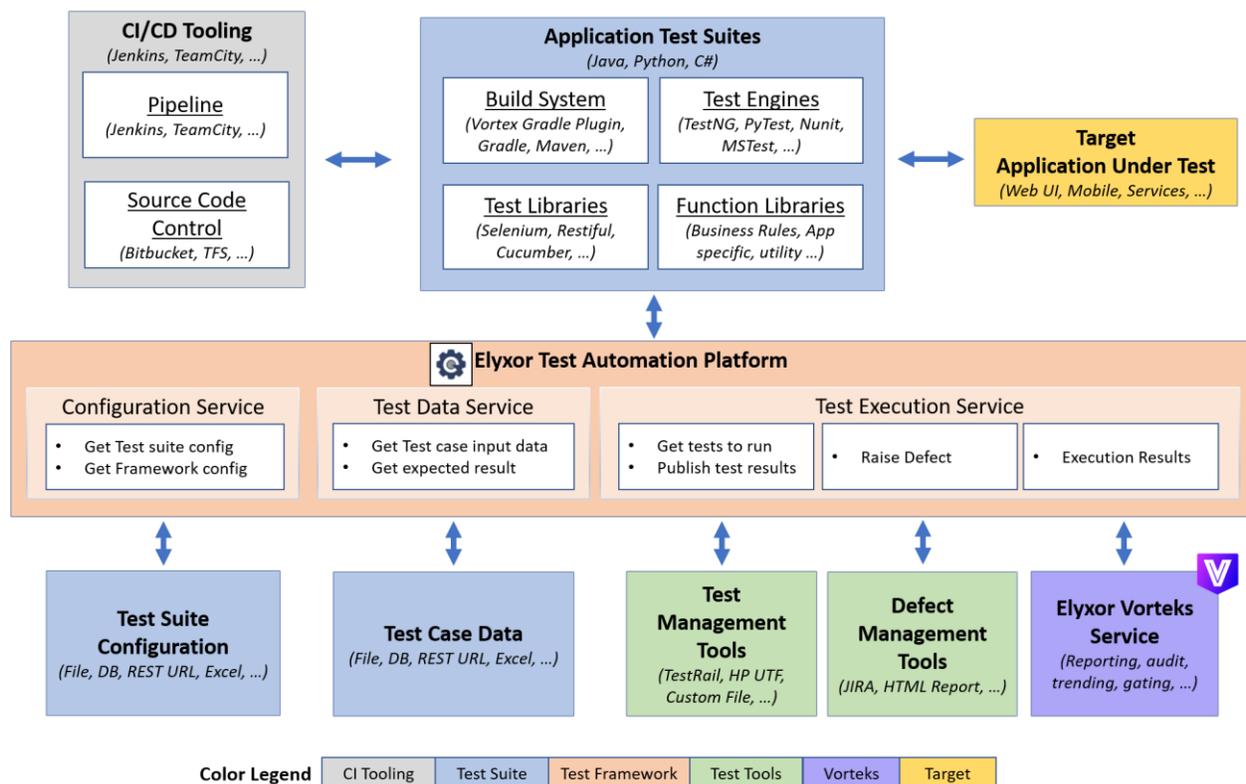


Figure 1. Test automation framework integration

ETAP can be leveraged by multiple test suites, enabling re-use and standardization.

Figure 2 depicts an example of a layered deployment design with the following features:

1. The **ETAP** core framework is packaged within an “enterprise” layer. This enterprise layer, for example, would be configured to use the enterprise test management tool for all usages of **ETAP**.
2. A business unit can then consume this enterprise artifact and extend it with common libraries. For example, a business unit which uses its own defect management tool, can override the enterprise configuration to connect to their tool.
3. The “test suite” layer for a specific application is built. This test suite will leverage both the enterprise and business unit common configuration and shared libraries.
4. Other tests suites can also be created that reuse these business unit layers

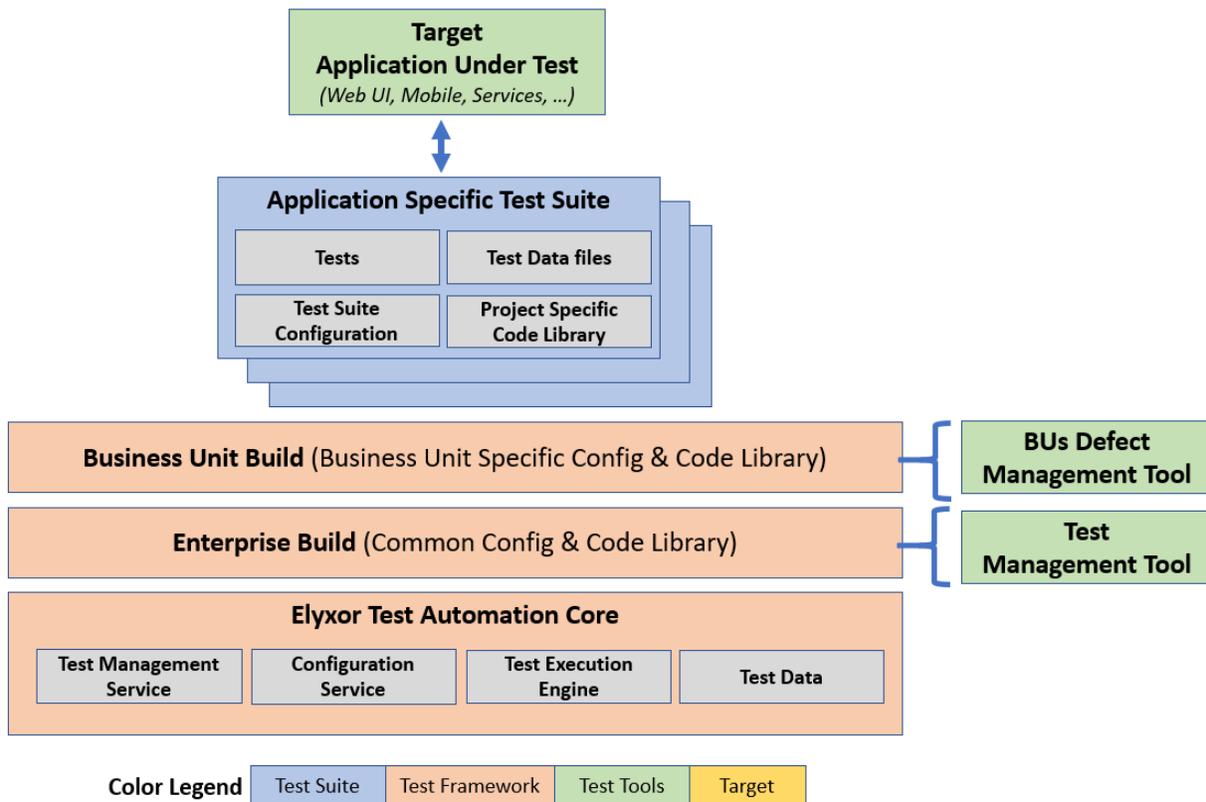


Figure 2. Example test automation framework implementation pattern

Results

ETAP, when in a **CI/CD** workflow, will result in the following benefits:

- Reduced risk related to deployed artifacts and configurations
- Increased stability of automated deployments.
- Faster time to delivery
- Eliminated manual activities
- Increased release velocity
- Increased speed to market
- Quicker return on investment

Considerations

- Define a test data strategy that supports continuous test execution and can be executed via automation
- Establish **CB** and **CD** for the environment and application under test (aim for a nightly build and automated test execution.)
- Define process and quality standards and enforce with a test management tool
- Design the test execution to support orchestration across application components
- Define a plan to have tests integrated into **CD** and treated as code
- Create requirements and an architecture that support testing, for example stubbed services

Vorteks

ETAP, in combination with the Elyxor Vorteks dashboard, provides visibility and controls to further improve the performance and productivity gains of any **CI/CD** implementation.

As shown in Figure 3, with Vorteks you will realize the following benefits;

- Full pipeline visibility
- Code and deployment auditing
- Quality policies enforcement
- Flexibility to support differing quality requirements
- Support for various technical implementations and data sources
- Scale from single project to Enterprise deployments

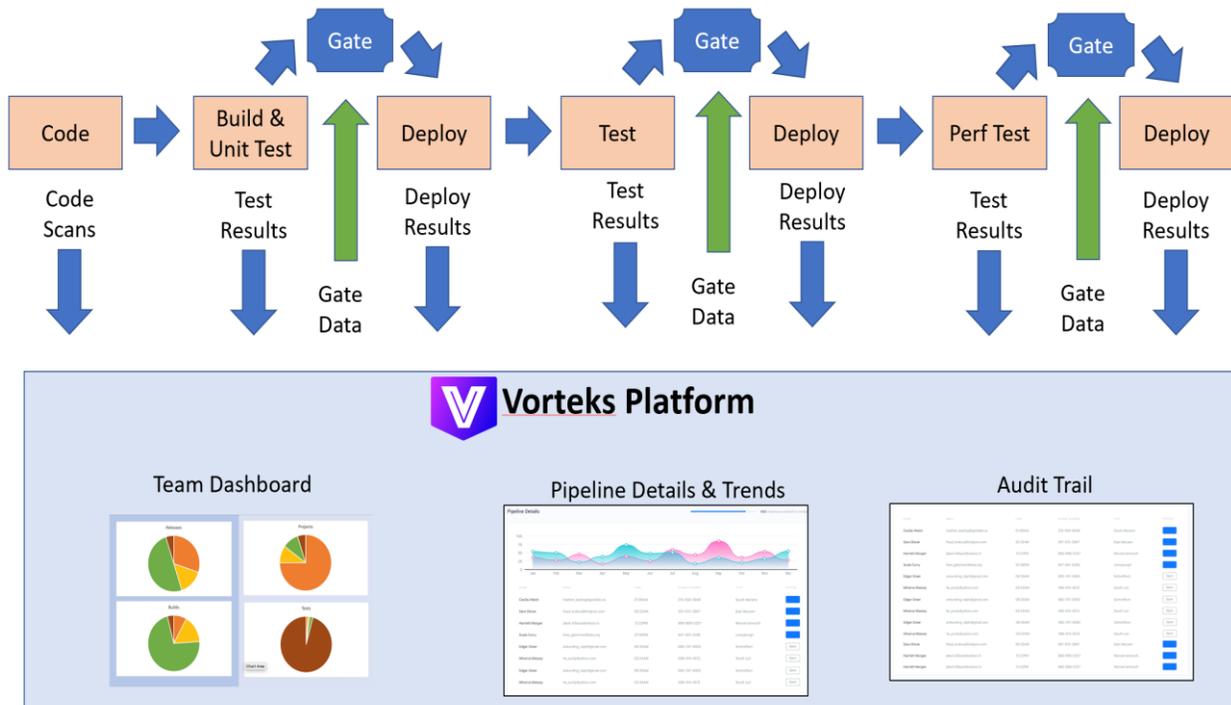


Figure 3. Elyxor Vorteks within the pipeline