

QA/Test Roadmap

'Barcelona': October 2017

'California': ~June 2018

'Delhi': ~October 2018

'Edinburgh': ~ April 2019

'Fuji': ~ October 2019

'Geneva': ~ April 2020

'Hanoi': ~ October 2020

Edinburgh Release (Committed)

- With automated blackbox tests in place, the Edinburgh release will include better visualization/dashboard of test results (including look up and display of historical test results). Candidate tools include Telegraf + Grafana + InfluxDB, DataDog, Prometheus. <https://github.com/edgexfoundry/blackbox-testing/issues/105>
- The Edinburgh release will include the capture of resource metrics to monitor performance. Metrics monitored will include memory and CPU consumption. <https://github.com/edgexfoundry/edgex-go/issues/115>
- As a stretch goal, the Edinburgh release will include performance/load testing using a tool like Bender, Jmeter, Load Impact or other tool selected by the work group.
- Security tests will be automated as mentioned above in the Security Group tasks. <https://github.com/edgexfoundry/security-api-gateway/issues/44> & <https://github.com/edgexfoundry/security-secret-store/issues/51>
- For the Edinburgh release, code contributors will now be required to supply additional or updated blackbox tests to cover changes made to the code base with any PR. Working group leads and project committers are head accountable for this change in procedures.
- As a stretch goal, the project will look to produce Swagger documentation to better support testing and API documentation in future releases. RAML documentation will remain the officially API standard for Edinburgh, but additional Swagger documentation will be provided and weighed for use in future releases.

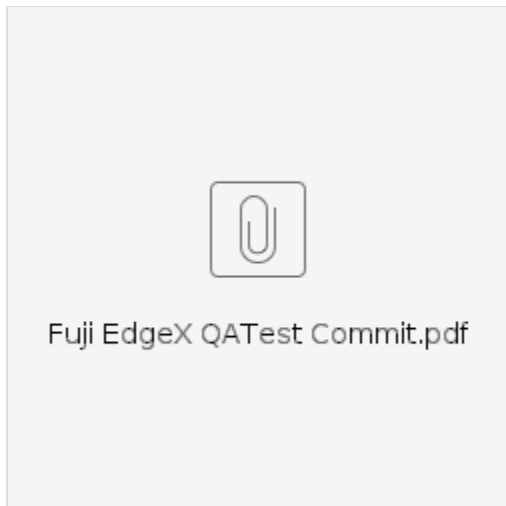
Backlog (April 2019)

Priorities shown in the table below are provisional and subject to review.

Item#	Description	Priority	Target Release
1	Improved unit tests across services (improvements should be driven by specific WG dev teams)	1	Fuji
2	Improve blackbox test structure including reorganization of the tests and better test case documentation (source ranya.ranganathan@intel.com)	1	Fuji
3	New test framework (e.g. Robot or Cucumber) to support additional types of functional/blackbox and system integration tests e.g. Device Service or system level latency tests	1	Fuji
4	Dockerize blackbox testing infrastructure for deployment simplification/flexibility	2	Geneva
5	System integration tests – currently we don't have any end to end tests e.g. Device Service read data -> Core Data -> Rules Engine or Export Service	2	Geneva
6	Blackbox tests for new Application Services microservices	1	Fuji
7	Blackbox test for Device Services - Virtual, Modbus, MQTT, BACNet, OPC UA. Develop common set of tests that can be used against all Device Services.	1	Fuji
8	Configuration testing – currently all blackbox tests run with single static configuration, we should test additional EdgeX configurations	2	Geneva
9	Automated performance testing - API Load testing (measure response time) and metrics (CPU, memory) collection for all EdgeX microservices (this work was started during the Edinburgh iteration)	1	Fuji
10	Automated performance testing - EdgeX microservice startup times (STRETCH GOAL FOR EDINBURGH)	1	Fuji
11	Automated performance testing - Automated system level latency and throughout testing (e.g. device read to export or device read to analytics to device actuation)	1	Fuji
12	Automated performance testing - Baseline performance of service binaries no container	3	Hanoi
13	Blackbox and performance test runs against other container technologies supported (e.g. snaps)	3	Hanoi
14	Automated performance testing - The ability to create summary reports/dashboards of key EdgeX performance indicators with alerts if thresholds have been exceeded	2	Geneva
15	Test coverage analysis e.g. using tools such as Codecov.io	1	Fuji

16	Static code analysis e.g. using tools such as SonarQube or Coverity to identify badly written code, memory leaks and security vulnerabilities	2	Geneva
17	Tracing during testing e.g. based on Open Tracing standard such as Zipkin, Jaeger	3	Hanoi
18	Replacement of RAML API documentation with Swagger	2	Geneva
19	Blackbox tests moved into edgex-go repository. Will allow blackbox test to be more easily added/updated by developers as part of a PR resulting from an API change.	3	Hanoi
20	Run blackbox tests for an individual EdgeX microservice when a PR issued.	3	Hanoi

Fuji Release (Committed)



- Improve blackbox test structure including reorganization of the tests and better test case documentation
- New test framework (e.g. Robot or Cucumber) to support additional types of functional/blackbox and system integration tests e.g. Device Service or system level latency tests
- Blackbox tests for new Application Services microservices. **Moved to Application Services WG**
- Blackbox test for Device Services - Virtual, Modbus, MQTT, BACNet, OPC UA. Initially develop common set of tests that can be used against all Device Services. **Moved to Device Services WG**
- Automated performance testing
 - API Load testing (measure response time) and metrics (CPU, memory) collection for all EdgeX microservices (this work was started during the Edinburgh iteration)
- Test coverage analysis e.g. using tools such as [Codecov.io](https://codecov.io). **Moved to Dev Ops WG**
- Replacement of RAML API documentation with Swagger

Geneva Planning

Backlog (November 2019)

Priorities shown in the table below are provisional and subject to review.

Item#	Description	Priority	Target Release
1	Improved unit tests across services (improvements should be driven by specific WG dev teams)	1	Fuji
2	Improve blackbox test structure including reorganization of the tests and better test case documentation (source ramya.ranganathan@intel.com)	1	Fuji
3	New test framework (e.g. Robot or Cucumber) to support additional types of functional/blackbox and system integration tests e.g. Device Service or system level latency tests	1	Fuji
4	Dockerize blackbox testing infrastructure for deployment simplification/flexibility	2	Geneva
5	System integration tests – currently we don't have any end to end tests e.g. Device Service read data -> Core Data -> Rules Engine or Export Service	1	Geneva
6	Blackbox tests for new Application Services microservices	1	Geneva
7	Blackbox test for Device Services - Virtual, Modbus, MQTT, BACNet, OPC UA. Develop common set of tests that can be used against all Device Services.	1	Geneva

8	Configuration testing – currently all blackbox tests run with single static configuration, we should test additional EdgeX configurations	2	Geneva
9	Automated performance testing - API Load testing (measure response time) and metrics (CPU, memory) collection for all EdgeX microservices (this work was started during the Edinburgh iteration)	2	Geneva
10	Automated performance testing – EdgeX microservice startup times (STRETCH GOAL FOR EDINBURGH)	1	Fuji
11	Automated performance testing – Automated system level latency and throughout testing (e.g. device read to export or device read to analytics to device actuation)	1	Fuji
12	Automated performance testing – Baseline performance of service binaries no container	3	Hanoi
13	Blackbox and performance test runs against other container technologies supported (e.g. snaps)	3	Hanoi
14	Automated performance testing - The ability to create summary reports/dashboards of key EdgeX performance indicators with alerts if thresholds have been exceeded	2	Geneva
15	Test coverage analysis e.g. using tools such as Codecov.io	1	Fuji
16	Static code analysis e.g. using tools such as SonarQube or Coverity to identify badly written code, memory leaks and security vulnerabilities	2	Geneva
17	Tracing during testing e.g. based on Open Tracing standard such as Zipkin, Jaeger	3	Hanoi
18	Replacement of RAML API documentation with Swagger	1	Geneva
19	Blackbox tests moved into edgex-go repository. Will allow blackbox test to be more easily added/updated by developers as part of a PR resulting from an API change.	3	Hanoi
20	Run blackbox tests for an individual EdgeX microservice when a PR issued.	3	Hanoi
21	Edinburgh/Fuji backward compatibility testing	2	Geneva