Project Incubation Exit Criteria

Introduction

The EdgeX Foundry Project has defined a lifecycle for its projects but this definition does not specify in any detail what it takes for a project to be able to transition from the *Incubation* phase to the *Active* phase.

This document is meant to fill this gap by defining a set of criteria to be considered before moving a project from *Incubation* to *Active*.

It is important to note that *Active* in this case refers to the project itself rather than its product and it is therefore more about the maturity of process than the maturity of the product or General Availability (GA).

For this reason this document defines two sets of exit criteria. The first one is made of requirements expected to be met by all projects before moving to *Active*. The second set lists examples of additional requirements typically defined at the onset of the project as goals to be met to exit incubation. There are expected to be documented in the Proposal for an EdgeX Project. Because not all projects have the same goals, the importance of each criteria and the exact definition of this second set of criteria may vary from one project to another. Ultimately the TSC is responsible for determining whether a project deserves to move to *Active* or not and this decision does not need to be solely based on these exit criteria. The purpose of these exit criteria is to help the TSC in its decision process by informing its members of key aspects of the project.

Minimum requirements

- Legal
    - All code has been made available under the Apache License and is free of incompatible dependencies
    - All documentation is made available under Creative Commons
    - Project name has been checked for trademark issues
- Community support
    - The project must have an active and diverse set of contributing members representing various constituencies
    - The project is not highly dependent on any single contributor (there are at least 3 legally independent committers and there is no single company or entity that is vital to the success of the project)
    - Release plans are developed and executed in public by the community.
- Sufficient test coverage
  The project must include a comprehensive unit and integration test suite and document its coverage. Additional performance and scale test capability is desirable.
- Sufficient user documentation
  The project must including enough documentation for anyone to test or deploy any of the modules.
- Alignment
    - Requirements fulfillment
      The project must document what requirements and use cases it addresses.
    - Architecture
      The project must document how it fits within the EdgeX Architecture
    - Compatibility with other EdgeX projects
      Where applicable, the project should be compatible with other active projects.
    - Release numbering: the project should use the EdgeX standard release taxonomy, once that is agreed upon.
    - Project must make a release, even a “developer preview”, before graduation.
- Infrastructure
    - Github repo has been created
    - Mailing lists have been created and are archived
    - Other communication means used, such as rocket.chat channels, are set up
    - Project is set up with Continuous Integration
    - All information necessary for someone to join the community and be able to start contributing is duly documented (location of repo, list of maintainers, mailing lists addresses, rocket.chat channels if used, etc) following the EdgeX Foundry Project standard practice (CONTRIBUTING.md MAINTAINERS.txt etc)

Additional considerations

In addition to the above, requirements such as the following may be defined at the onset of the project and considered as goals to be met to exit incubation:

- Sufficient real world use
  The project should be used in real applications and not just in demos. Because not all real applications may be discussed publicly, in such cases statements providing as much details as possible should be made.
- Sufficient scalability
  The project must demonstrate sufficient scalability and document its scalability over various dimensions such as:
    - Minimum latency between collection of sensor data to export, and from command being applied based on analytics
    - Minimum footprint for a single instance
    - Ability to scale to communications between many nodes in tiered deployments
    - Ability to support communications "north, south, east and west" in a distributed computing environment
- Minimum test code coverage expected, such as, for example:
    - Minimum coverage for Security & cryptography

    - Minimum coverage for other functionality
    - Minimum coverage for Business logic/smart contract

Acknowledgements

The above leverages process utilized by the Hyperledger Project, which borrows from the ASF's Minimum Graduation Requirements.