Project Readiness

NOTE: This is a rough draft and work in progress, feel free to add/modify the content.

Introduction

Project readiness is the general term we use to describe all of the things that the members of a project must do to make the project resilient to the ebb and flow of users, contributors, maintainers, and sponsors. A project that is ready can survive the loss of key maintainers and/or corporate sponsors. It includes many goals, achievements, and best practices in technical, cultural, educational, and business areas.

Maturing the Maturation Process

The project lifecycle outlines the different stages of a Hyperledger project and the transition from each stage to the next is governed by the technical steering committee (TSC). So far only the exit criteria for moving from incubation to active status have been defined. The exit criteria for the other transitions and the first major release have yet to be formally defined and blessed as governing criteria. Below is a list of the things we can do to mature this process to take the guesswork out of it.

- Define the exit criteria for each stage and the first major release.
- Create forms for each stage exit similar to the core infrastructure initiative application.
- Include some, if not all, of the CII gold criteria in the stages where they make sense.

Thoughts on Production Ready

For a project to be considered production ready there are some minimum infrastructure requirements that should be added to the exit criteria.

- CII badge application complete.
- Using Hyperledger infrastructure for
    - Wiki
    - Chat
    - Mailing list
    - Meetings
    - Bug tracking
    - CI/CD
    - Analytics

Measuring Readiness

Achieving project readiness requires that we gather analytics in many different areas primarily so that we can check to see if our efforts are resulting in improvements to the anti-fragitility and readiness. We have yet to define the entire list of measurements we should take when evaluating the readiness of the project. The areas where we should measure are below.

- People
    - breadth of contributor and maintainer corp
    - user to contributor conversion
    - contributor to maintainer conversion
- Code
    - contribution rates
    - bug rates
    - security defects
    - milestone schedules
- Marketing
    - reach
    - mentions
    - sponsorship
- Events
    - attendance
    - contributor conversion

Take Two –

Introduction and Purpose

Trying to describe concisely what it means for a project to be "ready" seems simple at first but there are so many details and layers of nuance that only a systematic breakdown is constructive. Establishing a solid idea for "being ready" can serve as the basis that guides the rest of this document. In the context of open source, being ready can be defined as having sufficient infrastructure along technical, cultural, educational, and business lines that a project becomes self-reinforcing and anti-fragile. Having quality, useful software developed using sound technical processes while nurturing a culture of openness and support motivates people to take ownership leading to self-reinforcing norms of continual improvement. Projects become anti-fragile when people genuinely care and then demonstrate that dedication following good cultural and technical ideals.

The purpose of this document is to capture everything we have done so far to promote project readiness, as well as all of the plans we have made, and attempt to synthesize a grand unified theory of readiness. To build a comprehensive picture we need to break down readiness into its constituent parts, define them, and then list how good practices in each part interacts with the others to make a project ready. The rest of this document examines the different roles and expectations of project participants and how they change with a project's maturity. Then the contribution of analytics to readiness is described and how it enables continual improvement. And lastly, each stage of a project's lifecycle is described with links to the already blessed exit criteria and expectations for a project at each stage.

Project Roles

Every project is a community and every community is made up of an aggregate of people playing one or more roles. It is important to acknowledge each role and to better understand the needs and desires of each one so that the community can both accommodate them and potentially convert them from roles with lower investment to ones with a higher investment in the interest of anti-fragility. Below is a list of roles that play the largest parts of an open source project community. It is not comprehensive but intended to address the major players.

Users

The users of a piece of open source represent the spring from which we draw all other kinds of people in the community. Contributors, maintainers, sponsors, advocates and educators all start as users. Users have the lowest level of investment an typically the lowest level of involvement but they have the highest level of potential for contributing to the future of a project. Their needs are paramount because if nobody uses a piece of software the community that creates and maintains it will evaporate and eventually disappear altogether. Thankfully the needs of users are simple. They only want open, reliable, and useful software that is both easy to learn and easy to use. Some would include that the software be customizable and/or modifiable as a user need but I see that as more of a feature that may support usefulness rather than a standalone need. The bottom line is, the goal of any open source project is to produce software that is useful and reliable.

Contributors

Contributors to an open source project is sometimes defined as anybody that has made any effort to support the project no matter how small. In this document, the definition is much narrower and is limited to anybody that has submitted a code or documentation change that is tracked. This does not exclude others that contribute to the project because there are categories below that cover people who make other forms of contributions. This narrow definition is congruent with the TSC approved definition for the purposes of deciding who can and cannot vote in the TSC elections. That measurement is also used to some degree when measuring project contributor health as well. The section on analytics discusses how we measure contributors.

Maintainers

Maintainers are a special class of contributor. They are the technical and cultural leaders of a project. They are trusted members of a project and are responsible for merging code changes, making official builds, and closing bugs as they are resolved. Maintainers are also responsible for setting the tone and culture for a project. The openness and welcoming nature of a project is largely set by the maintainers and the example they set for the rest of the participants. Another key responsibility of maintainers is to delegate responsibilities to contributors in an effort to convert them from contributors to maintainers. This is the primary means by which the maintainership can grow beyond the original core set of developers and make a project resilient to the ebb and flow of maintainers.

Sponsors

This includes any organization that contributes money and/or labor hours to a project. Sponsors are responsible for providing feedback on the overall direction of the project in conjunction with the maintainers. The resources that sponsor provide typically supports all of the consumable resources such as computer time, data storage, marketing, and support staff.

Advocates / Educators

Anybody that contributes to a project through other media such as writing, speaking, and creating educational programs/certifications fall into this category. They drive the public credibility of a project and do the most to grow the pool of users. Advocates often become sponsors and educators often become contributors and sometimes maintainers.

Analytics and Readiness

- Checking our work is crucial to continual improvement.
- What can we measure?
- What should we measure?
- What do we measure now?
- Post-mortems
- Bug bounties

Project Lifecycle

- The stages as defined so far.
- The existing blessed documents and where they land in the lifecycle
- Evaluating readiness at each stage.
- First major release.
- Subsequent major releases.

Every open source project follows an arc from inception to retirement. The stages that the TSC has identified and classified are: lab

Lab

Incubation

Active

Deprecated

Take One –

A project readiness plan encompasses all of the things that a Hyperledger project should be accomplishing to be ready for use in production environments. The Project Incubation Exit Criteria document covers a lot of a project requires to be "ready". The areas of focus break down into three rough groups of requirements: technical, community and business.

Technical

- open coding
- open bug tracking
    - good first bugs
- security
    - responsible disclosure
    - outside audits
    - code review policy
    - code style policy
    - test coverage
    - CI/CD
    - secure software delivery
- enterprise grade deployment artifacts
- documentation

Community

- Community Building Plans
- Project Lifecycle plan
- open meetings
- mailing list/chat
- archive of community discussions
- meetups
- hackathons

Business

- open source license
- marketing plan
- educational/training materials
- requirements gathering process