Xen.org Proposed Project Governance

Published on May 11, 2011, version 1.0

Problem Statement

Xen.org over the year hosted and today is hosting a number of projects. Today the following projects are highlighted as top-level Xen.org projects

- **Xen Hypervisor** - Leading open source hypervisor for servers … Learn More
- **Xen Cloud Platform** - Enterprise cloud solution stack for virtualization … Learn More
- **XCI** - Xen hypervisor for client devices … Learn More
- **HXEN** - Xen running as a Type 2 hypervisor … Learn More
- **Xen ARM** - Port of Xen hypervisor to ARM processor … Learn More
- **Project Satori** - Support for paravirtualized Xen Linux guests running on Hyper-V … Learn More

However these projects are in different states:

- Xen HV is mature, well and active and follows strong unwritten governance conventions
- XCP is in an early state, publishing its first release recently. XCP shows some signs of maturity, but in other areas is finding its feet. The project has big potential to build a community and become a key component of open source cloud stacks.
- XCI is in an early state and has not shown much momentum recently.
- HXEN does not have an active developer community and there has not been
- Satori is completed

What is the Proposal aiming to achieve

- A formally agreed governance will lead to more consistently run projects.
- Clear governance sets expectations and makes it easier for large vendors to engage with Xen.org
- Clearly positioned projects avoid confusion in the community, e.g.
  - Projects in early phases (=incubation) need more help to succeed and positioning projects as such increases the likelihood of getting such help
  - Clearly positioning projects into mature projects and project in early phases (=incubation) sets expectations: in other words people don't expect features that are developed in an incubation project to be mature. A concrete example where we have an (avoidable) issue is that XCI is in an early stage, but not clearly positioned as such. What we have seen as a consequence is that some users assume that Xen has functionality similar to VirtualBox and are disappointed and turn against xen.org when they realize the support isn't there.
- Improved project oversight, support and heartbeat ensure that projects gain momentum more quickly and can become successful more quickly. This would have an overall positive effect for Xen.org and its community.
- Setting expectations regarding governance makes it more attractive for projects related to Xen.org to associate with Xen.org rather than go elsewhere. Again this would have an overall positive effect for Xen.org and its community.
- The barrier of entry to creating Xen.org projects is undefined. Typically in open source this can encourages projects being created for the wrong reasons. The worst

case example is a project being created because “this sounds like a good idea”. The risk of failure for projects being created without clear intentions and support is quite high and can damage the reputation of an open source organisation.

Most open source communities solve this through a project life cycle that is consistently applied to projects. However there is a risk of creating bureaucracy, which can stifle the project. Examples of project life cycles can be found in Eclipse and Apache, but both are too heavyweight.

Goals

- Create a set of minimum requirements for a project
- Create a lightweight project life cycle that
  - leads the project to articulate its goals and how to achieve them
  - encourages desired behaviours (e.g. open development)
  - provides motivation for the project to succeed
  - leads to non-viable projects failing quickly
  - provides opportunities for other community member
- Avoid bureaucracy, i.e. the life cycle should be as informal as possible
- Encourage Xen related projects to be hosted on Xen.org rather than going elsewhere
- Set clear expectations to vendors, upstream and downstream projects and community members

The Proposal

Principles & Roles

The Xen Hypervisor project follows a number of unwritten conventions. We propose that these conventions should apply to all Xen.org projects, with a number of small additions and clarifications.

Principle 1: Openness

Xen.org is open to all and provides the same opportunity to all. Everyone participates with the same rules. There are no rules to exclude any potential contributors which include, of course, direct competitors in the marketplace.

Principle 2: Transparency

Project discussions, minutes, deliberations, project plans, plans for new features, and other artefacts are open, public, and easily accessible.

Principle 3: Meritocracy

Xen.org is a meritocracy. The more you contribute the more responsibility you will earn. Leadership roles in Xen are also merit-based and earned by peer acclaim.

Principle 4: Consensus Decision Making

Xen.org projects are normally auto-governing and driven by the people who volunteer for the job. This functions well for most cases. When more formal decision making and coordination is required, decisions are taken with a lazy consensus approach: a few positive votes with no negative vote are enough to get going.
Voting is done with numbers:

- +1 : a positive vote
- 0 : abstain, have no opinion
- -1 : a negative vote

A negative vote should include an alternative proposal or a detailed explanation of the reasons for the negative vote. The project community then tries to gather consensus on an alternative proposal that resolves the issue. In the great majority of cases, the concerns leading to the negative vote can be addressed.

Role: Maintainer

Maintainers are allowed to commit changes into the source code repository and their prime responsibility is to accept, co-ordinate and apply patches from other community member's and maintain the design cohesion of their components. In Xen.org maintainers own a set of components for which they are ultimately responsible for. Maintainers are at least listed in a MAINTAINERS file in the root of the source tree.

Role: Project Lead

Xen.org projects are managed by a Project Lead, who also is a maintainer. Project Leads are the public figure-head of the project and is overall responsible for the health of the project. The project lead typically also has write access

Proposal: Electing Maintainers

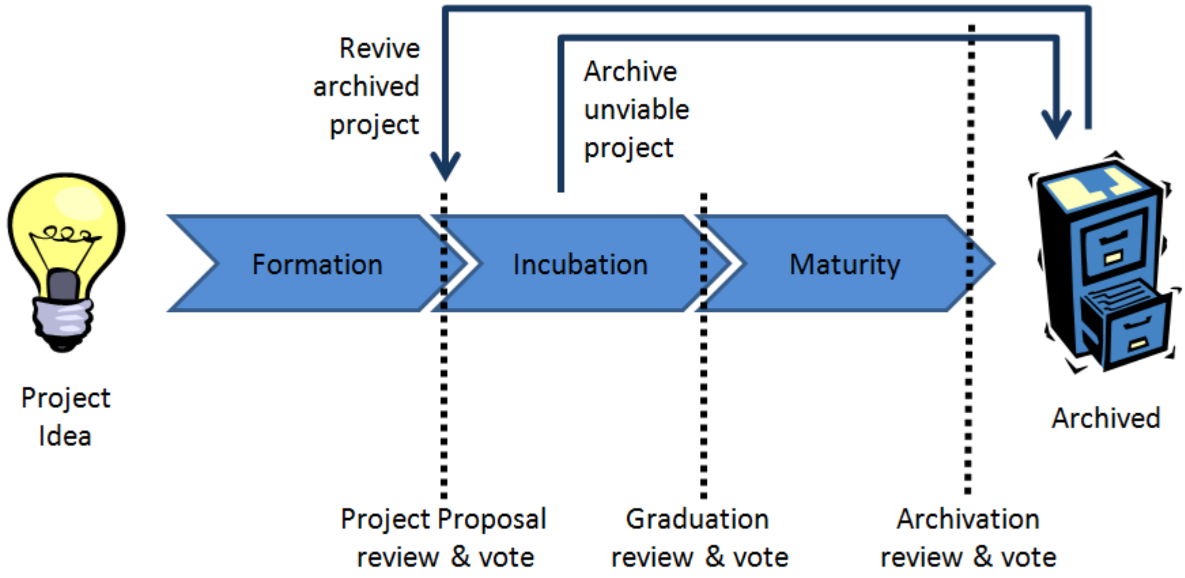
Xen.org currently has no formal way of determining new maintainers. The proposal is that developers who have earned the trust of maintainers (including the project lead) can through election be promoted to Maintainer. We propose a two stage mechanism

1) Nomination: A maintainer should nominate a community member explaining the candidate’s contributions to the project and thus why they should be elected as a maintainer on the project’s public mailing list. The nomination should include a scope (set of components). Cite evidence such as patches and other contributions where the case is not obvious.

2) Election: A maintainer will be elected using the decision making process outlined earlier. Voting should be done by maintainers for that project privately sending the vote to votes@xen.org or fill out an on-line form (e.g. Google docs form or similar). Should there be a negative vote the project lead and community manager will try and resolve the situation and reach consensus. Results will be published on the public mailing list.

Basic Project Life Cycle

The proposal is to follow a simple basic flow:



A project starts with an idea which through the process of project formation will grow into a project proposal. The project proposal will need to satisfy some basic conditions, will be put out for community review and is then put to a vote to all maintainers and project leads of **mature** Xen.org projects following the usual decision making process..

For agreed project proposals Xen.org will provide basic infrastructure and the project can get started. Projects in incubation are working towards a set of goals, will get additional support and marketing opportunities from Xen.org. However there will also be regular checkpoints to see whether the project is progressing. Should it turn out that a project is not viable any more, it will be archived after an archivation review and vote. For a project to graduate, some basic conditions must be satisfied. If a project in incubation has achieved the point where it believes it is mature enough to graduate, it can request a Graduation community review followed by a vote.

Mature projects are pretty much expected to run themselves. However at some point a mature project will lose momentum and developers. If this is the case the Xen.org community can request an archivation review, which follows the usual pattern. Archivation reviews have two purposes: a) give somebody in the community an opportunity to step up and continue a project, b) archive the project outcomes such that they are still available to people who want to use them, but promotion of such projects will cease.

It is also possible to revive archived projects. However these are treated almost like new projects as projects would only be archived if they have become inactive.

How this works in Practice

In this section, we will explore how the project life cycle works in practice. The Xen.org community manager will act as facilitator

Requesting Reviews, Reviews and Voting

Requesting Reviews: Project Proposal and Graduation Reviews are requested by the (prospective) project lead of the project by contacting the community manager providing the necessary documentation. An archivation review can be requested by any maintainer of a mature project or by the Xen.org community manager. The community manager will then publish relevant material on the respective mailing lists.

Reviews: These are off-line reviews which are open to all community members by which a proposal is published for review. The purpose of the review is two-fold: a) gather final feedback and input from the community (it is good practice to informally do this before the review), b) advertise the project with the aim to attract interest, users and contributors. After a review, the requester of the review may decide to withdraw, request a re-review or progress to a vote by arranging with the community manager.

Voting: The community manager arranges a timed private vote made to votes@xen.org (voting should be open for a minimum of a week). Any maintainer of a mature project and the Xen.org community manager is allowed to vote. Voting follows the conventions as laid out in *"Principle: Consensus Decision Making".*

Forming a Project

Requirements for forming a Xen.org project:

- A project needs a lead, who is willing to become the project lead of the project
- A project needs a sponsor, which can be a project lead of a mature project, a member of the Xen.org advisory board or the community manager
- A project needs a mentor, which can be the project sponsor or a maintainer of a mature project
- A project needs to have a relationship to other Xen.org projects, i.e. it aims to develop software that has a dependency on other Xen.org projects. If the project needs components in other Xen.org projects to work, then this should also be stated.
- A project needs to be large and long-term enough to grant a separate project. For example adding support for a new CPU architecture, adding additional functionality on top of existing projects, etc. Adding a new feature to an existing project should be performed within an existing project.
- A project will deliver code using a license that is compatible with other Xen.org projects (ideally GPLv2).

The purpose of the project formation phase is to work out what the project is about, get community buy-in and help the future project gain publicity and momentum. The formation phase is driven by the project lead. The project mentor's role is to advise and support the project lead in getting the project started.

The project proposal is a document that describes and is published on wiki.xen.org:

- What the project is aiming to achieve (i.e. the project charter and project goals)
- What components/code and in which code lines (new or components in other projects) the project aims to deliver
- Key dependencies on other Xen.org projects (if applicable)
- Lists initial maintainers (if applicable)
- Lists any interested parties in the project (if applicable)
- Lists any planned initial code contributions (if applicable)
- A rough plan on how to get through the Incubation phase

Project Proposal Review

The review is initiated by the project lead and follows the rules outlined in *"Requesting Reviews, Reviews and Voting"*

After a successful review, the following resources will be created for the project:

- A mailing list
- A codeline
- A project portal on Xen.org (in an area separate from mature projects)
- A wiki page on Wiki.Xen.org (this is expected to be maintained by the project lead)

Incubating a Project

The purpose of the incubation phase is for a project to show that it is gathering momentum and adheres to the “Principles & Roles” of Xen.org projects. The project mentor will work closely with the project lead and there are at least quarterly informal review meetings with the mentor on how the project is doing. Should a mentor not be able to fulfil his/her role any more, it is the responsibility of the project lead to find another mentor. We advise that the project lead gives at least quarterly updates on the Xen.org blog on how the project is doing.

Xen.org will provide support to incubating projects. The project lead will work closely with the Xen.org community manager as well as with the project mentor.

Archiving an Incubating project:

The mentor can request for a project to be archived, if the project is not making sufficient progress. See *“archivation review”*.

Graduation Review

The review is initiated by the project lead and follows the rules outlined in *“Requesting Reviews, Reviews and Voting”*. In essence the project lead makes a pitch to the community, why the project should graduate.

A project must fulfil the following requirements before it can graduate:

- It follows the principles of openness, transparency and meritocracy
- It has delivered at least one functioning release of what it is aiming to deliver
- It has a public code line which shows active development and has mechanisms to accept patches (and a history of accepting patches)
- It has a public mailing list that is active (as we get more experience we will add some guidelines)
- It has a mechanism for users to raise bugs and for developers to work on bugs
- It has an active developer community (as we get more experience we will add some guidelines). But things to look for are number of maintainers, different organisations involved, number of users, etc.

Other items to look at during the review (depending on project are):

- It has an up-to-date wiki and a core and group of people maintaining it
- It publishes regular builds and tests
- It promotes itself at events and on the blog
- The way how the project prepares for public releases

Mature Projects

Mature projects are expected to be run and promote themselves. The project lead has significant responsibility in ensuring that this happens. Xen.org and the community manager will help organize events, provide opportunities for the project to get new contributors and build a community, promote new releases on the blog and to the press, work with project members, etc. However Xen.org and the community manager will not get involved in the day-to-day running of the project.

At some point during its life cycle a project may lose momentum. In other words developers and users are not interested in the project any more. If this is the case, it may be time to archive the project. If the project has achieved its goals and is thus completed, it may also be time to archive the project.

Archivation Review

These can happen in a number of situations:

- An incubation project shows clear signs of failing and not progressing
- A mature project has lost its developer and user base (and there is little or no activity)
- The project has achieved its goals and/or fulfilled its charter: in other words it has completed

In the first case the review is triggered by the incubation project's mentor. Failing this the review can be requested by any maintainer of a mature project (including the project's lead) or by the Xen.org community manager. See *"Requesting Reviews, Reviews and Voting"*.

The review is essentially a pitch why the project should be archived. The purpose of the review is not necessarily to archive a project, but also to provide a last opportunity for interested parties in the Xen.org community to save the project and step up. The Xen.org community manager will support efforts to save the project, should community members want to step up. There is the special case that a project has been completed: in this case the normal situation would be for the project lead to make the case, why this is so.

Archived Projects

When a project is archived the following happens:

- The codeline and mailing list will be made read-only and made accessible from an archived projects section on Xen.org
- The project's wiki pages will be tagged as "archived". A project may be completed (i.e. it has achieved its goals and/or fulfilled its charter) in which case it is tagged as "completed" and "archived".
- The project portal on Xen.org will be moved into an "Archive" section. We may have a "Completed" section within the "Archive" section.

In cases where the project has delivered code into other Xen.org projects (e.g. IA64 support in xen-unstable), the code will be

- Deprecated at the point where the project is archived
- The project which now contains the archived code can (but does not have to) remove the code in a subsequent release (it should however give users sufficient time to adapt)

Exceptional Circumstances

Projects without project lead

Projects which lose their project lead during the incubation or maturity phase are at risk of failing. Should this occur, the project's maintainer community should agree who would want to be/be able to be the new project lead and follow the election process as outlined in *"Electing Maintainers"*.

If a project lead leaves during the formation phase, without finding a successor we assume that the project does not have enough momentum and will not go ahead.

Incubation projects without mentor

Should an incubation project lose its mentor, the Xen.org community manager will support the project lead in finding a new mentor.

What would this mean for Xen projects today?

The following table assesses individual projects on Xen.org and proposes action to be taken. It is clear that for existing projects, where there is no obvious case, we have a number of choices:

- Classify the project according to the new process in agreement with stake-holders
- Only apply the process to NEW projects
- Agree on a migration route with stake-holder towards the process for projects where the situation is not clear

The ideal approach would be to apply the process to ALL projects where the situation is clear and find a migration route where it is not.

Xen Hypervisor	**Mature Project**
XCP	**Incubation or Mature Project** The project shows some of the signs of a mature project, however does have a few localized issues around momentum and maturity. This is expected for a project at this point. The project only just delivered its 1.0 release in March 2011. From a process perspective, one could argue that the project is mature BUT it could be beneficial to place it just before reaching maturity and graduate the project (as a test case for the process).
XCI	**Incubation Project** In its early stages (no published builds, no releases) and has not built much momentum yet. Project goals may benefit from a review.
Xen IA64	**Candidate to be Archived** Has some code in xen-unstable.hg, which is broken. There is no project lead and no active community. We have contacted people on the xen-ia64-devel mailing list to see whether there are any active IA64 developers, and give the community an opportunity to step up. The result is that there do not appear to be active developers.
HXen	**Candidate to be Archived** No project lead, no active developer community for >1 year.
Satori	**Candidate to be Archived** No project lead, no active developer community for >1 year. The project actually has completed.
