

How the ASF works!

Stefano Mazzocchi
<stefano@apache.org>

This session will give you everything you always wanted to know about the foundation but were afraid to ask. The difference between membership and committership, who decides what, how elections take place, how is our infrastructure setup, what is the board, what is a PMC, what's the philosophy behind the incubator, why is the foundation moving away from project containment. Come and see behind the scenes of the ASF.

The Apache Software Foundation (ASF) is a non-profit organization incorporated in the United States of America and was formed primarily to:

- provide a foundation for open, collaborative software development projects by supplying hardware, communication, and business infrastructure;
- create an independent legal entity to which companies and individuals can donate resources and be assured that those resources will be used for the public benefit;
- provide a means for individual volunteers to be sheltered from legal suits directed at the Foundation's projects; and,
- protect the 'Apache' brand, as applied to its software products, from being abused by other organizations.

That's the dry fact, but how did all this come to be and what does it really mean in its details? We have to step back a little in history.

A little bit of history

The foundation was created in 1999 by a group of people that called themselves the "Apache Group" and that got together, several years earlier, to continue to support and maintain the HTTPD web server written by the NCSA.

That server was freely available, came with source code and was licensed under a license that allowed very open modification and redistribution, but the original developers lost interest in that project and moved onto something else, leaving users with no support.

Some of those users started to exchange fixes (friendly called “patches”) and information on how to prevent problems and improve the existing software. Brian Behlendorf created a mailing list on his own machine for those users to collaborate to fix, maintain and improve that software.

In fact, the legend says that the name “Apache Web Server” comes from “a patchy web server” and that group of users started to become developers and started to release their own version of the software, and started to call themselves the “Apache Group”.

Between 1995 and 1999, the Apache HTTPD Web Server created by the Apache Group became the leader of the market (and currently still is, with more than 60% of the web sites in the world powered by it).

But as the web grew bigger, economical interests started to grow, and the Apache web site hosting new sister projects (such as the mod_perl project, the PHP project, the Java Apache project), the need for a more coherent and structured organization that would shield individuals from potential legal attacks felt more and more necessary.

Meritocracy

Unlike other software development efforts done under an open source license, the Apache Web Server was not initiated by a single developer (for example, like the Linux Kernel, or the Perl/Python languages), but started as a diverse group of people that shared common interests and got to know each other by exchanging information, fixes and suggestions.

As the group started to develop their own version of the software, moving away from the NCSA version, more people were attracted and started to help out, first by sending little patches, or suggestions, or replying to email on the mail list, later by more important contributions.

When the group felt that the person had “earned” the merit to be part of the development community, she was granted direct access to the code repository, thus increasing the group and increasing the ability of the group to develop the program and maintain/develop it more effectively.

We call this basic principle “meritocracy”: literally, govern of merit.

What is interesting to note is that the process scaled very well without creating friction, because unlike in other situations where power is a scarce and conservative resource, in the apache group newcomers were seen as volunteers that wanted to help, rather than people that wanted to steal a position.

Being no conservative resource at stake (money, energy, time), the group was happy to have new people coming in and help, they were only filtering the people that they believed committed enough for the task and matched the human attitudes required to work well with others, especially in disagreement.

The chain of merit

There are different “merit stages” inside the ASF:

```
user -> committer -> member
```

where:

- a “user” is anybody that uses the software but doesn’t have write access to the code repository. She can be passive (doesn’t participate in mail list or discussions, also known as “lurker”) or active (participates in discussions, sends patches, documentation, suggestions, criticism, also known as “contributor”).
- a “committer” is an active user that was elected because of his merit for the evolution and progress of the project. She has write access to the code repository, an @apache.org mail address, the right to vote for the community-related decisions and the right to propose an active user for committership.
- a “member” is a committer that was elected because of his merit for the evolution and progress of the foundation. Legally, a member is the “shareholder” of the foundation, one of the owners. She gets the right to elect the board, to candidate herself for the board election and to propose a committer for membership. She also gets the right to propose a new project for incubation (we’ll see later what this means)

The foundation structure

As the Apache Web Server started to grow in market share and popularity, due to synergy of its technical merit and to the openness of the community behind the project, people started to create satellite projects. Influenced by the spirit of the community they were used to, they adopted the same traditions of community management.

So, by the time the ASF was created, there were several separate communities, each focused on a different side of the “web serving” problem, but all united by a common set of goals and a respected set of cultural traditions in both etiquette and process.

These separate communities were referred to as “projects” and while similar, each of them exhibited little differences that made them special.

In order to reduce friction and allow for diversity to emerge, rather than forcing a monoculture from the top, the foundation was created with a hierarchical structure and self-government of the various parts (the projects) was encouraged, by allowing each project to create its own technical charter and its own government rules.

At the same time, the cultural influence of the original apache group was strong and the similarities between the various communities are evident, as we'll see later.

The structure of the foundation is hierarchical:

foundation → project → sub-project

where:

- foundation is the entire ASF, contains one or more projects
- project is the unit of self governance of the foundation, contains one or more sub-projects
- sub-project is a group that focuses on a particular common effort (and it's normally associated to a software product)

The foundation is governed by the following entities:

- the Board of Directors (board) governs the foundation and is composed of members.
- the Project Management Committees (PMC) govern the projects and they are composed of committers. (note that every member is, by definition, also a committer)

The Board of Directors (board)

The board is responsible for management and oversight of the business and affairs of the corporation in accordance with the foundation Bylaws. This includes management of the corporate assets (funds, intellectual property, trademarks, and support equipment) and allocation of corporate resources to project

However, technical decision-making authority regarding the content and direction of the Apache projects is assigned to each respective project management committee.

The board is currently composed by nine individuals, elected between the members of the foundation. The bylaws don't specify the number of officers that the board should have, but historically, this was the number of the first board and it never changed.

The board is elected every year.

The Project Management Committees (PMC)

In addition to the officers of the corporation, the Board of Directors establishes, by resolution, Project Management Committees consisting of at least one officer of the corporation, who shall be designated chairman of such committee, and may include one or more other members.

Each Project Management Committee is responsible for the active management of one or more communities identified by resolution of the Board of Directors.

Subject to the direction of the Board of Directors, the chairman of each Project Management Committee is primarily responsible for the communities managed by such committee, and she has the power to establish rules and procedures for the day to day management of communities for which the committee is responsible, including the composition of the PMC itself.

The board has the faculty to terminate a PMC at any time by resolution.

Project management

It must be noted that since the appointed PMC has the power to create its own self-governing rules, there is no single vision on how PMCs should run a project and the communities they host.

At the same time, while there are some differences, there are a number of similarities shared by all the projects:

Communication

Communication is done via mailing lists. These identify “virtual meeting rooms” where conversations happen asynchronously, which is a general requirement for groups that are so geographically distributed to cover all time zones (like it’s normally the case for the various apache communities)

Some projects use more synchronous messaging (for example, IRC or instant messaging). Voice communication is extremely rare, normally because of costs and the language barrier (speech is harder to understand than written text).

In general, asynchronous communication is much more important because it allows archives to be created and it’s more tolerant on the volunteer nature of the various communities.

Decision Making

Projects are normally auto governing and driven by the people who volunteer for the job. This is sometimes referred to as “do-ocracy”, power of those who do and works for most cases.

When coordination is required, decisions are taken with a lazy consensus approach: a few positive votes with no negative vote is enough to get going.

Voting is done with numbers:

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

Etiquette suggests that a negative vote includes an alternative proposal or a detailed explanation on the concerns for the negative vote.

The community then tries to gather consensus on an alternative proposal that resolve the issue. In the great majority of cases, the negative vote is removed or turned into abstain.

This process is called “consensus gathering” and we consider it a very important indication of a healthy community.

In those rare cases when the -1 cannot be resolved, a majority vote can be called and the negative vote overruled, but each project has its own details on how this happens, but healthy and diverse communities tend to avoid majority ruling to remove friction because an overruled -1 is the first step toward forking friction.

Philosophy

While there is not an official list, these six principles have been cited as the core beliefs of philosophy behind the foundation, which is normally referred to as “The Apache Way”:

- collaborative software development
- commercial-friendly standard license
- consistently high quality software
- respectful, honest, technical-based interaction
- faithful implementation of standards
- security as a mandatory feature

All of the apache projects share these principles.

Operation

All projects are composed of volunteers and nobody (not even members or officers) are paid directly by the foundation for their job. There are many examples of committers that are paid to work on the projects, but never by the foundation themselves, but rather by companies or institutions that use the software and want to enhance it or maintain it.

The foundation infrastructure

The ASF does not have offices or buildings, it’s a virtual entity that exists only on the internet. It’s only physical existence is the technical infrastructure that runs it.

The infrastructure of the foundation is mostly composed of the following services:

- the web serving environment (web sites and wikis)
- the code repositories

- the mail management environment
- the issue/bug tracking
- the distribution mirroring system

The ASF has an “infrastructure” team that is responsible for the management and day-to-day system administration and operativity of the various hardware that runs the above services. Some of the members of the infrastructure team volunteer to be “on call” and be paged in case the system goes down.

The infrastructure team is also responsible for approving the installation of a new system or software on the ASF machines.

The foundation incubator

In order for new projects to be created, the ASF created a project called Incubator which is responsible to help new efforts to join the foundation.

Since the meritocratic rules work from top to bottom, it is vital, for the long-term stability of such a form of government, that the initial set of committers has to know very well the dynamics of such a system as well as share the same philosophical attitude toward collaboration and openness that the ASF expects from its projects.

The incubator is responsible for:

- filtering the proposals about the creation of a new project or sub-project
- help the creation of the project and the infrastructure that it needs to operate
- supervise and mentor the incubated community in order for them to reach an open meritocratic environment
- evaluate the maturity of the incubated project, either promoting it to official project/sub-project status or by retiring it, in case of failure.

It must be noted that the incubator (just like the board) does not perform filtering on the basis of technical. This is because the foundation respects and suggests variety of technical approaches. It doesn't fear innovation or even internal confrontation between projects overlapping in functionality.

The incubator filters projects on the basis of the likeliness of them becoming successful meritocratic communities and the basic requirements for incubation are:

- a working codebase – over the years and after several failures, the foundation came to understand that without an initial working codebase, it is generally hard to bootstrap a community. This is because leadership is not well recognized by developers without a working codebase. Also, the friction that is developed during the initial design stage is likely to fragment the community;

- the intention to donate copyright of the software and the intellectual property that it may contain to the foundation – this allows the foundation to obtain an irrevocable and permanent right to redistribute and work on the code, without fearing lock-in for itself or for its users;
- a sponsoring ASF member or officer – this person will act as the main Shepherd, giving directions to the project, helping out in the day-to-day details and keeping contact with the incubator PMC.

The incubation period normally serves to estimate whether or not:

- the project is able to increase the diversity of its committer base and to play with the meritocratic rules of the foundation.

It might seem rather easy to achieve, but it must be remembered that in a volunteer and highly selective environment, attracting new committers is not automatic.

Diversity of committership is important for two main reasons:

- it gives long term stability to the project development – in fact, with all the developers affiliated to the same entity, the chance of seeing all of them moving away from the project at the same time is much greater than with a community of individuals affiliated to unrelated entities;
- it gives a greater variety of technical visions – something that guarantees a better adherence to environment and user’s needs, thus a higher change of finding real-life use of the software.

Other foundation entities

After infrastructure and incubator, the foundation hosts several other entities more or less formalized open to ASF members and to invited experts or individuals that do not directly create code but serve for specific purposes. They are:

- the *conference organizing committee* (aka *concom*) – responsible for the organization of the official ASF conference (aka ApacheCon);
- the *security committee* – responsible for the handling of potential security holes in the software produced by the foundation that might impact our users. It gets contacted by the finders of the problems before the problem report is made available to the public, to allow the projects to provide a fix in time for the report, thus reducing vulnerability to a minimum;
- the *fund raising committee* – responsible for the fund raising (collaborates with the *concom* since the conference is one of the major source of income of the foundation)
- the *JCP committee* - responsible for the liaison between the ASF and the Java Community Process (of which Executive Committee, the ASF is member);

- the *licensing committee* – responsible for the legal issues associated with licensing and license compatibilities and for the revision of the Apache Software License;
- the *press committee* – responsible for the press-related issues like press releases for major ASF events or dispatching requests for interviews.

The ASF also hosts a few foundation-wide mailing lists:

- *announcement* – is a moderated mailing list where announcements of releases or major ASF events are made
- *community* – is the ASF-wide mail list, readable by all but writable only by ASF committers. It is mostly used to discuss issues that touch that entire foundation.

Conclusions

With 4 years of operation, the ASF represents one of the best examples of an open organization that was able to find balance between structure and flexibility. It grew from 200 of committers to almost 800 and exhibits no trends of saturation. It was able to create several software products that are leaders in their market, competing and winning with commercial closed-sourced alternatives. It was able to find balance between openness and economical feasibility, earning respect from a range of people, from single individuals, to multinational corporations and providing inspiration for businesses, governments, education and for other software foundations.