

# Apache server - A successful open-source project

Dexter Drupsteen  
5887224

January 27, 2013

## 1 Apache

The Apache HTTP server is one of the most widely used http servers available on the net. Since its appearance in 1995, the server has only grown in popularity. Nowadays more than half of the websites is hosted by an Apache server.

Apache is developed by an open source community under the control of the Apache Group or, since 1999, known as the Apache Software Foundation, a distributed network of core software developers. The Apache Software Foundation is a non profit corporation, founded to provide legal protection to contributors and making sure the Apache name is not used in vain.

Under the second generation Apache Software Foundation there are a lot of development groups, hereinafter Apache Groups or AGs. The developers in the Apache groups are all volunteers and have earned their credits in the Apache community by being experts on the whole software package or a lead developer on a smaller part of one of the the Apache projects. Core developers that have lowered their interest in their Apache project, or who do not have enough time during a long period can be removed from an AG. Thus the membership of an AG is not a lifetime membership and fully based on the quality and quantity of contributions of a developer. Between the initial release of Apache (1995) and 2000 there were at least twelve core developers member of the AG (the Apache HTTP Server core development group) and an additional 3 developers being in the race to join the AG. Nowadays the core development group of the Apache HTTP Server consists of 35 members.

It is clear that the Apache HTTP Server is a success. It is the http server on the web. But why did the development of Apache HTTP Server succeeded? How is the development coordinated and how does development on this FOSS project compares to commercial projects?

## 2 Apache HTTP Server

What is different in style when comparing commercial projects to open source project? First of all, the project is developed by hundreds of developers. These developers are not managed in anyways and pick up the work they would like to do. And when they pick up the development work, there is no global system-level design at hand for them to converge to. And to put the cherry on top of the cake, there is no real project plan, schedule or backlog kind of list. All the

work done is pure voluntary and the quality of the code is unknown. But, as with every project done by volunteers, some do more than others. Those who do some more than the rest of the community, and do this for a period of time are alligable to join the core development group.

As said, the core development group for Apache HTTP Server Consisted of 8 to 15 members between 1995 and 2000. These core developers are responsible for producing eighty percent of the code in production. We can see that these developers are experts on the matter. Most of the developers in a core group are users themselves of the software they create. Because of that, they have a total clear perspective on how the software should work, and which functionalities are missing. But not only are they writing a lot of code for the Apache HTTP Server project, they are also responsible for releasing new versions of the Apache project and reviewing patches that have been written by the community.

In the period we are discussing, almost 400 developers worked on the Apache HTTP Server project. And as said, 15 developers were responsible for 80 percent of the code, that means that 320 developers are responsible for around 20 percent of the code. Most of these changes or features are reviewed by a few people before being made into patches for the core development group to accept or reject. Note that the 320 developers are not all professionals or experts on the Apache HTTP server, but apparently they are users of the software, why else would you contribute.

On top of the 400 individuals that contribute to the project with lines of code, there is a group consisting of an astonishing 3000 individuals that contribute to the project by writing problem reports (i.e. reporting bugs, non working functionalities or anomalies in the software). Of those 3000 individuals, 2600 problem reporters reported a problem that not resulted in changes in the code or to the project. All these problem reports are not checked by the core developers, but rather volunteers with somewhat more knowledge than the problem reporter. These developers either reject the problem by removing them from the system, or forward the problems to the main mailing list for the core developers. To summarize the work of a developer in the Apache project: find out there is a problem; find solutions for the problem (for instance by discussing it on the mailing lists); check if there is a volunteer to do the fix (or do it yourself); review the fix(es) and check if there is a core developer to put it into the next release.

### 3 Commercial comparison

When comparing the productivity of the developers of Apache with commercial developers (guys who actually get paid for the work they do), statistics show us that the top 15 Apache developers are at least as productive as their commercial colleagues, even more so if one takes into account that the Apache developers do this part time.[1] Also because the development of the Apache HTTP Server is based on consensus within the core development team, we can see that developers on the core team are less likely to make mistakes or defects in their programming. This because the code is reviewed multiple times by peers in the community, and solutions are openly discussed in the newsgroups or the mailing lists. The fact that a lot of contributors write code for several modules within the project (a form of qualitative code ownership, i.e. ownership over the code if

one thinks one has a bit of expertise on the subject) also helps this, as the code goes across a lot of developers, all with different backgrounds, coding abilities and ideas about solutions.

But with less bugs released into production by the community, how is the time to resolve problems in the project, and how are they prioritized? The strange thing is that, as the developers are users, thus the core developers are users as well, that priority is a matter of perspective. The priority given to a problem report by a bug reporter never corresponds to the priority within the community or the core team. The priority is usually based on the number of users (thus developers) affected or the component affected. Bugs or problems in the core system (i.e. kernel, protocol or vital parts of the system) are dealt with first; secondly the most wide spread bugs are taken care of followed by major optional and operating system depending bugs (and although the priority of documentation varies enormously, it is mostly fixed sooner than major optional and OS depending bugs, due to the fact that especially in open source software documentation is of vital importance).

## 4 Conclusion

It is a beautiful fact that the most successful http server in the world is an open source project. The fact that motivated, globally distributed volunteers work together towards a better piece of software and succeed in that with creating less problems than commercial colleagues is astonishing. Even when wanted features, the problems, architecture and a, by management created, vision are missing. Though, it is a shame that only fifteen developers contributed eighty percent of the code. A lot more could be done when the remaining 320 developers had an equal amount of productivity. One of the problems there lies in the fact that there are not enough resources in the community devoted to find and fix bugs and problems in the code done by an even larger group (the fifteen developers are supported by the rest of the community as they find, and fix the bugs in the code the core developers have written).

The commercial development can certainly learn from the open source world. The flexibility of the open source communities and the concept of peer reviewing are things that can easily be incorporated in commercial software development. Although this might be impractical and on certain points impossible, it would be a real improvement if commercial top developers also were top users and really understood the product as a normal user would do. And in the end, the user of both commercial or open source software should realize that he is the most valuable source of feedback. Open source software nor commercial software can really improve without the continuous feedback of the people who use it.

## References

- [1] A Case Study of Open Source Software Development: The Apache Server, Mockus, Audris, Fielding Roy, and Herbsleb James, Proceedings of the International Conference on Software Engineering (ICSE 2000)
- [2] Netcraft Market Share for Top Servers Across All Domains, August 1995 - November 2009