Maintaining, Preserving, and Distributing Open-Source Software for the Classics

The Perseids Project [www.perseids.org/] provides an editing and annotation environment for working with ancient texts, as well as a suite of tools for learning and teaching the classics. One of the major challenges faced by the project is the preservation of its digital tools. Unlike physical texts, software requires constant maintenance in order to continue to run and provide useful services to end-users. Due to constraints in the digital humanities, including the necessity of subject-matter expertise, budget limitations, and the small number of users compared to commercial applications, maintaining software and tools becomes a difficult challenge. This paper will explore how developers in the digital humanities can write software with preservation in mind, how a more open development model could benefit the project, and how we are addressing these issues with Perseids.

First, the paper will discuss how software can be partially future-proofed by writing it defensively: using well-known libraries, following standards, and conforming with best practices. It is often said in programming circles that software developers should "use the right tool for the job" [Ambler 2003]. When it comes to development of software for the classics, deciding on the "right tool" must include thinking about preservation for the future, not simply language features or the ease of development for the present.

This paper will also discuss how the "bazaar" model of development of open-source projects, as described in the Eric S. Raymond's 2000 essay, *The Cathedral and the Bazaar*, applies to the Perseids Project and digital classics software in general. There is a community of passionate amateur and self-taught ancient language enthusiasts who are familiar with technology and application development, but the barrier to entry on many projects is very high. By making the project more open and welcoming, both technically and non-technically, developers outside of academia can be encouraged to get involved. This will make software more likely to be maintained because it allows the community to continue working on software that is no longer supported by an institution. It will also result in higher-quality software, since, in open-source projects, “given enough eyeballs, all bugs are shallow” [Raymond 2000].

Finally, this paper will give examples of what is being done at the Perseids Project to address these issues. It will discuss how maintenance and contribution are balanced with ease-of-use and functionality when writing software for use in the classics.

Bibliography

Ambler, Scott. 2003. “The Right Tool for the Job.” Website: *Dr. Dobb's: The World of Software Development*. http://www.drdobbs.com/the-right-tool-for-the-job/184415067. Accessed August 15, 2018.

Raymond, Eric Steven. 2000. “The Cathedral and the Bazaar.” http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/. Accessed August 15, 2018.