Managing Knowledge Within Military Open Source Software (MOSS) Projects

Pierre Barbaroux and Thomas Le Texier

Research Center of the French Air Force
Defense and Knowledge Management Laboratory

pierre.barbaroux@inet.air.defense.gouv.fr

Abstract. The Open Source Software (OSS) development model is arousing increasing interest from both practitioners and theorists. OSS models are congenial for both private companies and public administrations since they provide software users and suppliers with highly responsive and innovative development models which are likely to reduce costs and to provide additional autonomy within critical technological domains. OSS are characterized by the distribution of source code to any developer interested in contributing to the provision of the code. License terms allow a developer to make unlimited copies of the source code which can be modified indefinitely by developers to fulfil particular needs. Each developer is then free to distribute her work and exploit it commercially depending on license terms. The U.S. Department of Defense (DoD) is currently introducing Open Source Software (OSS) development models through its Open Technology Development (OTD) project (Herz et al. 2006). OSS adoption is becoming popular for military purposes. Indeed, the digital transformation of the U.S. military has generated a strong need for new modes of software management. In this context, the OTD project tends to set up an innovative model for software acquisition, maintenance, updating, and renewing. Focusing on the adoption of Military Open Source Software (MOSS) models by the U.S. Department of Defense (DoD), this article suggests that the open nature of OSS development platforms induces significant organizational changes related to the acquisition, transformation and creation of a variety of OSS core capabilities. Furthermore, we suggest that the U.S. military organizations should perceive modularity in software architecture as a means to establish effective modes of knowledge management and innovation. Building on the distinction between architectural and component knowledge, we find that modular source code structuring facilitates the division, distribution and coordination of both architectural knowledge and component knowledge within heterogeneous OSS developers’ communities.