

Open Source software in medical informatics--why, how and what

presented by Patrick Herron on 23 March 2004

Clement J. McDonald, Gunther Schadow, Michael Barnes, Paul Dexter, J. Marc Overhage, Burke Mamlin and J. Michael McCoy, Open Source software in medical informatics--why, how and what, International Journal of Medical Informatics, Volume 69, Issues 2-3, March 2003, Pages 175-184

McDonald *et al* discusses the status of open source software and open standards in medical informatics in an attempt to advocate their use in the field.

The paper defines the history of the open source movement and details how licensing provides the basis for open source software. The merits of open source software, the authors argue, add numerous advantages over commercially licensed software, including reduced cost, peer review, "shallow bugs," dynamic development, corporate independence, niche development, innovation, room for the little guy, and greater security.

McDonald *et al* reminds us that open source is nothing new, even in medicine. Open standards, the authors maintain, are as essential to the development of successful medical information systems, as they provide interoperability and stability along with the built-in benefit of multiorganizational cooperation and agreement. HL7 and ICD9 are as essential as open standards to medical informatics as standards like HTML and JPEG are to the Internet.

The authors cite a number of examples of current and successful medical open source projects, including OSCAR, OpenGALEN, and the authors' own 3rd generation Regenstrief Medical Record System.

The authors conclude their evangelical paper by suggesting how the use of open source in medical informatics might be accelerated. They recommend that a Federal policy be created to require that all federally supported medical software development projects be produced with open source licensing, with open standards, and layered upon previous open source projects. McDonald *et al* conclude that the final and most important part of establishing open source and open standards in medical informatics is with evangelism, with open proselytization of open source.

McDonald *et al* is unabashedly evangelical and in essence puts its own recommendation in action. In a certain sense the authors' job is made easy: it is indeed difficult to argue that open source has *not* been essential to the development of core information technologies, or that it might not be beneficial to medical informatics in the foreseeable future. In another sense, such an argument is absolutely necessary given that specialized domains not typically in the center of information technology, such as the field of health care, are rarely well-informed about open source. The authors succeed in reinforcing the compelling nature of open source.

In order for the paper to be more convincing it might have been appropriate for the authors to engage the concept of adoption; in particular the paper might have addressed the reasons behind the fears of medical professionals driving them away from eagerly fostering and adopting open source solutions. Being even slightly critical of open source might serve to address some well-known concerns about open source.

Another concert was raised in the class discussion. The paper seemed to muddle the concepts of open source with open standards, which are different concepts that share only a sentiment and a word. The general idea of openness is present in both, but the similarities end there: standards provide defined parameters on doing things and exchanging things (e.g., SQL) while open source regards openly licensed software with modifiable code. Discussing both open source and open standards, however, do serve the ultimate purpose of the paper: to advocate good practices for medical software development.

Class discussion time was expended on expounding on the history of open source as well as the advantageousness of the "shallow bug:" how open source invites broad code scrutiny and in so doing makes each bug tractable.

A final criticism of the paper: it may be hard to mandate code layering in Federal grants, in enforcing when software development projects incorporate code from other projects. It may be reasonable to advocate reuse practices but enforcing them is not practical if not occasionally inappropriate.