Talking about OSS: Making Sense of the Bazaar: 1st Workshop on Open Source Software Engineering – Workshop Report

J. Feller, B. Fitzgerald and A. van der Hoek

1 Introduction

Making Sense of the Bazaar: 1st Workshop on Open Source Software Engineering was held on 15th May 2001, in Toronto, Ontario, Canada, as part of The 23rd International Conference on Software Engineering (ICSE 2001). The goal of the workshop was to bring together interested researchers and practitioners to discuss, and better understand, the perhaps unique dynamics that are involved in Open Source Software Engineering. There were 30 participants, representing universities, companies, research centres and government organisations in Canada, Ireland, Germany, Japan, the Netherlands, Sweden, the United Kingdom and the United States (a list of participants is available online at http://opensource.ucc.ie/icse2001/participants.htm).

As noted in the introduction to this special issue, since the coining of the term Open Source Software (OSS) in 1998, there has been a steady rise in research on the topic. This is not particularly surprising; many Open Source products (the Apache HTTP server, the BIND DNS implementation, etc.) are category leaders in the Internet application space, and others (like Linux, NetBSD, etc.) are becoming increasingly popular as components in enterprise computing architectures. Perhaps more importantly, Open Source has become a major part of the strategic plans of traditional software giants like IBM, Sun, Netscape, Oracle and others. The somewhat unexpected commercial interest in OSS, and the demonstrated quality of OSS products, calls for a deeper understanding of the processes and tools that support it. The increase in Open Source research has not lessened our need for more accurate and rigorous analysis of the phenomenon. If anything, the research to date has raised more questions than it has answered. Thus, our goal in organising the workshop was two fold—to encourage the exchange of ideas between researchers and practitioners in this area, and to help shape the rapidly evolving Open Source research agenda (the formal workshop goals can be found at http://opensource.ucc.ie/icse2001/CFP.htm).

2 Position papers

The 30 workshop participants, both alone and in collaboration, submitted 17 position papers, which were published in the printed workshop proceedings and are available in full online at http://opensource.ucc.ie/icse2001/papers.htm. The 17 papers are described briefly below.

Several of the position papers examined the question of ‘how’ Open Source Software Engineering takes place. For example, in ‘Leveraging open-source communities to improve the quality and performance of open-source software’, Douglas C. Schmidt and Adam Porter discussed the effectiveness of OSS processes in reducing life-cycles and decreasing development and quality assurance costs for certain types of software. At the same time they drew attention to the challenge presented by OSS—long term maintenance and evolution, end-user confidence, coherency and usability. Ulf Asklund and Lars Bendix, whose work on configuration management in OSS appears in this issue, attended the workshop and presented the roots of their ideas in a paper entitled ‘Configuration management for open source software’. Marcus Bittman, Robert Roos and Gregory M. Kapfhammer, in ‘Creating a free, dependable software engineering environment for building Java applications’, brought to the workshop some initial findings from a study that ‘took stock’ of the OSS development platform available to OSS researchers. Although they found several powerful tools, they also reported a lack of analysis and design tools, as well as a lack of tools focused on developing the user interface. While Bittman et al. addressed the coding tool space, ‘The ramp-up challenge in open-source software projects’, by Davor Cubranic, examined a different type of developmental artefact—namely the communication and collaboration that takes place within OSS communities. Finally, Yoshiyuki Nishinaka, in ‘Open source software developments in XP style’, contributed an introspective look at three OSS products with which he had been involved (XME, Tcl/Tk Japanization Kit, and Jun) and discussed the connections between these OSS projects and the tools and techniques of Extreme Programming (XP).
Another group of papers focused more on the question of 'why' OSS takes place - issues of motivation and commitment. In 'Open source development: an Arthurian legend', Jonathon E. Cook presented a somewhat light-hearted account of the heroes, villains and benevolent dictators so often visible in the OSS world, and proposed the more serious question of how talented developers are attracted and retained by OSS projects. 'Reputation layers for open-source development', by Hassan Masum, picks up on this theme as well, seeking to identify practical solutions to the problems of managing reputation and rewards in OSS. Finally, Walt Scacchi presented a working account of the research that eventually led to the paper that appears in this special issue, in 'Software development practices in open software development communities: a comparative case study'.

The third group of papers all looked at the application of OSS beyond OSS, both now and in the future. In 'Corporate source: applying open source concepts to a corporate environment', Jamie Dinkelacker and Pankaj K. Garg discuss the Hewlett-Packard experience with 'Corporate Source', an effort to replicate OSS dynamics within a traditional organisational setting. Jai Asundi, in 'Software engineering lessons from open source projects' addressed the same question on a more theoretical level, offering general suggestions on aspects of OSS development that were 'portable' to more traditional contexts. Dan Port and Gail Kaiser discussed their experiences with using OSS methods to improve the learning experience of Software Engineering students in 'Introducing a “street fair” open source practice within project based software engineering courses' - yet another port of OSS to a new platform. Finally, in this section, 'Open source development: a suitable method to introduce a standardized communication protocol?' by Achim Spangler narrowed the focus of conversation by looking at the suitability of OSS for a very specific context - the development of a reference implementation of two international agricultural communication protocols.

If the papers above looked to some extent at the potential 'outputs' of OSS in various spaces, the remaining papers looked toward the 'inputs' - theoretical frameworks and experiences that help to contextualise OSS. Kumiyo Nakakoji and Yashushi Yamamoto, in 'Taxonomy of open source software development', highlighted the heterogeneous nature of OSS and presented interesting insights into how the goals of different development groups shaped the particular 'brand' of OSS that they practiced. Echoing the novel metaphor of Jonathan E. Cook's paper, Kouichi Kishida's paper, 'Conceptual sociological model for open source software' discussed the underpinnings of OSS in the context of the 'free cities' of the Mongolian empire. Scott A. Hissam and Charles B. Weinstock, whose work appears in this issue, reversed the perspective of Dinkelacker and Garg in their paper 'Open source software: the other commercial software'. Rather than raise the question, 'What can closed source software learn from OSS?' they asked 'What can OSS learn from our experience in using traditional commercial off-the-shelf products (COTS)?' Last, but certainly not least, 'Software architecture and open source software - where can research leverage the most?' by Budi Arief, Cristina Gacek and Tony Lawrie, and 'Software engineering research in the bazaar' by Ahmed E. Hassan, Michael W. Godfrey and Richard C. Holt, both addressed the issues of software architecture - and the lessons to be learned by OSS developers from the years of industry experience in this area.

### 3 Presentations and discussion

The workshop consisted of four sessions, each 90 minutes long. In each session, the author(s) of one of four selected position papers was asked to deliver a brief presentation, which was followed by open discussion. The presentations were as follows:

- **Session one**: Leveraging open-source communities to improve the quality and performance of open-source software (Schmidt and Porter).
- **Session two**: Open source development: an Arthurian legend (Cook).
- **Session three**: Corporate source: applying open source concepts to a corporate environment (Dinkelacker and Garg).
- **Session four**: Taxonomy of open source software development (Nakakojo and Yamamoto).

These four papers were invited for presentation because it was felt that they effectively drew out many of the key issues involved in the four themes discussed in the previous section - namely operational challenges (how), motivation and community (why), future and novel applications (whither) and informing frameworks (whence) - and thus well represented the papers as a whole. The presentation slides are available at [http://opensource.ucc.ie/icse2001/slides.htm](http://opensource.ucc.ie/icse2001/slides.htm).

Six hours is a surprisingly short period of time, and the day's discussion focused unsurprisingly on only a handful of issues. In the morning sessions, the majority of discussion revolved around the idea of OSS communities. Several key questions were discussed at length, including:

- How do you enable/empower non-developer users to make meaningful contributions to OSS projects?
- How reliant are OSS projects on, as Jon Cook put it, a handful of 'benevolent dictators'?
- How do you reconcile the highly personal 'scratch your own itch' motivation behind many OSS products, with the clear requirement of meeting the needs of other end users?

In the afternoon sessions, the focus of discussion shifted to more operational and technical areas. In particular, the participants exchanged ideas on:

- The relative complexity of application spaces in which OSS has a particularly strong record (servers, low level utilities, etc.), and the implied question of whether OSS was suitable for requirement/design-heavy spaces like payroll systems, ERP systems, etc.
- The true cost of what was once called 'free software' in developers' time, users time, knowledge sharing, etc.
- Whether OSS offered a context for truly rapid software evolution, or simply for rapid development and debugging.

All in all, the workshop was quite successful. It was a rewarding day of discussion, and served to help the participants better articulate the questions guiding their individual research agendas. More importantly, the workshop acted as a springboard for other work. Many of the papers have since evolved into full research papers (some of which are published in this special issue) and the workshop has led to several research collaborations between previously unrelated participants.

### 4 The 2nd workshop

*Meeting Challenges and Surviving Success: 2nd Workshop on Open Source Software Engineering* will be held on
Saturday, 25th May 2002 as part of The 24th International Conference on Software Engineering (ICSE 2002), in Buenos Aires, Argentina (date and venue subject to change). The goal of this year's workshop is to bring together researchers and practitioners to discuss both the short and long term sustainability of OSS. Specifically, the workshop will seek to address the quality and maintainability of OSS products, the replicability and portability of the OSS software engineering process, the stability and sustainability of OSS developer and user communities, and the viability and profitability of OSS business models. Further information on the 2nd workshop can be found at http://opensource.ucc.ie/icse2002.