Paper PO06-2009

Open Source SAS® Software Applications: the OS3A Program for Community Programming

Paul OldenKamp, Seattle Children's Research Institute, Seattle, WA, USA
Paul D. Hamilton, Amgen Corporation, Seattle, WA, USA
Dante DiTommaso, F. Hoffmann-La Roche AG, Basel, Switzerland
Ann Marie Martin, UCB Pharma, Braine l'Alleud, Belgium

ABSTRACT

The Web 'changes everything'!! We are entering a new paradigm of worldwide collaboration in scientific, educational and economic activity based on a large decrease in communication costs created by the Internet. This will change how we write programs in the future. The paradigm shift has made possible new methods of Open Source software development and distribution. Can SAS® software users take advantage of the opportunities?

How could this Open Source model ever work? At first glance, the Open Source model should fail as an example of the free rider problem of classic economics. The adverse incentives can be explained with a "programmers' dilemma" analysis in game theory. The Open Source licenses have conditions that overcome the free rider problem based on the principles of the Open Source Initiative.

As a practical matter, the facilities provided by several sites supporting Open Source projects and the practices developed by the many Open Source projects over the past 15 years are key in generating cooperative peer production by software application developers.

INTRODUCTION

Several authors have created a library of books and articles that present their vision of a new environment for software development based on community peer production. We will survey several references to provide the reader with motivation for the Open Source movement. There is a growing collection of economic theory and analysis of the legal and organizational characteristics that overcome the traditional public goods/free rider market failure conclusions. We will discuss the practical techniques used by Open Source projects and the specific implementation of an Open Source project for developing SAS applications. Finally we consider future directions for SAS developers interested in Open Source participation.

THE WEB 'CHANGES EVERYTHING'!!!

The Internet has resulted in revolutionary changes in the cost of communications and has made possible revolutionary new models of economic activity in the information economy. This revolution is occurring in many domains from political affairs and scientific research to peer production and consumer/producer firms. For SAS users not only will the Open Source revolution provide new complimentary systems to be used with SAS Software but methods of collaborative development and the exchange of programs through the Internet will open up new efficiencies in our work.

Below we will summarize a few of the sources that present these visionary projections. Some of these authors describe a broad transformation of society and economic production. Other authors focus more narrowly on the Open Source code programming of computer systems that is closer to our own work.

OPEN SOURCE TECHNOLOGY AND POLICY: SOFTWARE COMMODITIZATION – Deek and McHugh

> "The open source movement is a worldwide attempt to promote an open style of software development more aligned with the accepted intellectual style of science than the proprietary modes of invention that have been characteristic of modern business. The idea – or vision – is to keep the scientific advances created by software development openly available for everyone to understand and improve upon. Perhaps even more so than in the conventional scientific paradigm, the very process of creation in open source is highly transparent throughout. Its products and processes can be continuously, almost instantaneously scrutinized over the Internet, even retrospectively. Its peer review process is even more open than that of traditional science. But most of all: its discoveries are not kept secret and it lets anyone, anywhere, anytime free to build on its discoveries and creations."

WEALTH OF NETWORKS: HOW SOCIAL PRODUCTION TRANSFORMS MARKETS AND FREEDOM – Yochai Benkler

Benkler's book provides a scholarly theoretical and philosophical analysis of the 'networked information economy' from the perspective of economics.

> "We have an opportunity to change the way we create and exchange information, knowledge, and culture. By doing so, we can make the twenty-first century one that offers individuals greater autonomy, political communities greater democracy, and societies greater opportunities for cultural self-reflection and human connection. We can remove some of the transactional barriers to material opportunity, and improve the state of human development everywhere. Perhaps these changes will be the foundation of a true transformation toward more liberal and egalitarian societies. Perhaps they will merely improve, in well-defined but smaller ways, human life along each of these dimensions. That alone is more than enough to justify an embrace of the networked information economy by anyone who values human welfare, development, and freedom." (Benkler, p. 473)

Some of Benkler's economic analysis will be considered in more detail later in this paper.

WIKINOMICS: HOW MASS COLLABORATION CHANGES EVERYTHING – Tapscott and Williams

Tapscott and Williams also have a chapter that explains peer production with the examples of Open Source software development and new media styles typified by the Wikipedia:

> "…. [Peer production] is a way of producing goods and services that relies entirely on self-organizing, egalitarian communities of individuals who come together voluntarily to produce a shared outcome. In reality, peer production mixes elements of hierarchy and self-organization and relies on meritocratic principles of organization—i.e., the most skilled and experienced members of the community provide leadership and help integrate contributions from the community."
>
> "In many peer production communities, productive activities are voluntary and nonmonetary. They are voluntary in that people contribute to these communities because they want to and because they can. No one orders a worker to post an article to Wikipedia or to contribute code to the Linux operating system. They are nonmonetary because most participants don't get paid for their contributions (at least not directly), and individuals determine if, what and how much they want to produce. Just because people don't get paid to participate in peering does not mean, however, that they do not benefit from their participation in other ways. We'll come back to this point in a moment." (Tapscott and Williams, p. 67)
>
> "…. Peer production works because: the new economics unleashed by technology have permanently altered the costs and benefits of producing information and collaborating; it is more efficient than firms or markets at allocating time and attention for certain tasks; it is really good at attracting a more diverse, broadly dispersed talent pool than individual firms can muster; and contributors enjoy the freedom and experience of peer production. In short, peer production works because it can." (Tapscott and Williams, p. 71)

OPEN SOURCE PARADIGM SHIFT – Tim O'Reilly

The software industry is in the middle of a paradigm shift that O'Reilly compares to the 1981 introduction of the PC with off-the-shelf components that opened up the design to other manufacturers. He says that IBM did not understand the consequences of its decision in 1981 and that Open Source developers do not fully understand the consequences now. To help understand the shift we need to look at how three long term trends that are expressed in Open Source:

- The *commoditization* of software
- Network-enables *collaboration*
- Software *customizability* (software as a service)

<u>Commoditization</u> Like several other observers, O'Reilly describes the trend towards commoditization of software that is driven by standards and a communications-centric environment. But he also analyzes the effects of the commoditization, in that there is an apparent large drop in the value of software as measured by the market revenue it earns. While revenues can drop to near the marginal cost of zero, the social value to users of the software greater than before the commoditization.

The more important observation made by Clayton Christensen is that even if the opportunity for profits on the commoditized software disappears, new opportunities emerge in related activities such as software as a service and software support services that add value to the software.

<u>Collaboration</u> The cooperation and sharing of resources observed in Open Source projects does not date from Richard Stallman's formal specification of the General Public License and the formation of GNU project in the 1980's. The practice was typical in the early days of computing when hardware, not software, was the primary source value. The cooperation shown in development of Unix and the role of Usenet in the development of the Internet needs to be better acknowledged as part of the movement. The cooperative trend started in the early software industry has extended to some of the most interesting and successful Internet based applications. For example, the grid

applications like SETI@home, the NASA Mars mapping clickworkers projects, or the Amazon.com user reviews. This is leading us to a variety of peer production and customer involvement that is the thesis of Tapscott and Williams in Wikinomics.

Customizability One attribute of successful applications of software as a service on the Internet is adaptability. It is now recognized that the basic architecture of the software is vital in allowing groups of distributed individuals to work cooperatively. The architecture of the web is the premiere example of a system whose design leads to cooperation. The effect of the proper architecture is likely more important to the success of cooperative projects than volunteerism or economic incentives. The ease of customizing Internet application with tools like Perl, PHP, Python, and Ruby as well as higher level tools like Durpal, Joomia, and Plone is a key part of the architecture.

O'Reilly concludes, "Instead of thinking of open source only as a set of software licenses and associated software development practices, we do better to think of it as a field of scientific and economic inquiry, one with many historical precedents, and part of a broader social and economic story. We must understand the impact of such factors as standards and their effect on commoditization, system architecture and network effects, and the development practices associated with software as a service."

INTELLECTUAL PROPERTY AND OPEN SOURCE – Van Lindberg

Lindberg says that we need to know the "*why* and *what*" of Open Source before examining how it works. "At its core, open source is a *legal construct for cooperation and trade in intellectual property.*" He uses an analogy to commercial banks (a corporate model organization like traditional software companies) and credit unions (cooperative model organizations like Open Source projects) to examine several important characteristics of Open Source projects. The different ownership characteristics, corporate models have separate owners and consumers while in cooperative models the owners and consumers are the same. This leads to different incentives for involvement and investment in the production effort. Like for profit banks and non-profit credit unions the different profit motives of proprietary software development and Open Source projects lead to different levels of cooperation. ".... [T]he cooperative nature of open source software allows the users of the software to solve computing problems cooperatively. By pooling coding resources, the participants in the open source project receive the full benefits of the software product while only paying a fraction of the total cost."

The market profile of products has a complex relationship with the ownership and profit characteristics. Products can be distinguished as specialty, high-profit goods and commodity goods. Traditional software has depended on the specialty-good nature of software to support its proprietary closed model of development. Partially driven by the Open Source movement, many classes of software are becoming commodity goods without the profit margins Just as events in the world economy have overtaken choices previously available for economic policy so that 'We are all Keynesians now.", the trend for more low cost commodity software restricts the choices on producing high-margin specialty software. "We are all Open Source now" may be the situation in a few years.

"So why does open source work?" The cooperative nature lowers the cost of contributions by an individual in comparison with the value of the product produced. But Open Source development is unstable because participants have incentives to become free-riders and decrease cooperation. The role of Open Source licenses in overcoming this incentive is a core aspect.

HOW COULD THIS POSSIBLY WORK?

How can anyone make money developing Open Source software and then giving it away? How can teams of developers cooperate and contribute code over long periods of time when any individual can use the code without contributing? Developers have the incentive to be free-riders and do not contribute to the community.

Classic economic analysis predicts the underproduction of public goods, goods that can be consumed by one person without decreasing the quantity available for others. Since the marginal cost of more consumption is zero the market price of these public goods will approach zero over time. There will be no price incentive for producers and production will eventually drop to zero.

PRISONER'S DILEMMA BECOMES THE PROGRAMMER'S DILEMMA

The Programmer's Dilemma table shown below illustrates the free rider incentives. There is a corresponding table for Programmer B with the labels reversed. The society solution would be the sum of the two tables. The sub-optimization is seen because the best choice for Programmer A is to defect even though the best solution overall would be for both to cooperate.

Programmer's Dilemma – Outcomes for Programmer A

	Programmer B cooperates	**Programmer B defects**
Programmer A cooperates	Contribution cost (-5) Working application (+10) (+5) Overall	Contribution cost (-5) (-5) Overall
Programmer A defects	Appropriate code (+5) (+5) Overall	Full cost (-10) Working application (+10) (0) Overall

Source: Van Lindberg, p170

The table below has been modified with the penalty payment from Programmer A to Programmer B if A appropriates B's code and violates the Open Source license. The penalty modifies the incentive for Programmer A so that the best choice is to cooperate.

Open Source Licensing – Outcomes for Programmer A

	Programmer B cooperates	**Programmer B defects**
Programmer A cooperates	Contribution cost (-5) Working application (+10) (+5) Overall	Contribution cost (-5) Reimbursement for copyright violation (+7) (+2) Overall
Programmer A defects	Appropriate code (+5) Pay for copyright violation(-7) (-2) Overall	Full cost (-10) Working application (+10) (0) Overall

Source: Van Lindberg, p172

NEW TECHNOLOGY MAKES IT ALL POSSIBLE– Benkler

Yochai Benkler makes four important economic observations in his book **The Wealth of Networks**. First, the idea that proprietary intellectual property is essential in our economy is overstated. Second, the expansion of patent, copyright laws, and trademarks in recent years have subsidized proprietary information models and cost non-proprietary models. Third, basic changes in information technologies have made non-proprietary models of information production more attractive than they ever have been. And fourth, the rise of peer production has challenged the conventional wisdom on the economics of information production.

The new technology of the Internet has lowered the cost of capital resources need to produce and distribute information and this has created the opportunity for non-proprietary peer production efforts. This change is comparable to the change in information production that resulted from the invention of the printing press.

Benkler also expands the traditional two models of production, market based and firm based, with a third model of equal importance, social production. The characteristics that make social production work best in certain situations are explained.

OPEN SOURCE LICENSES AS ECONOMIC SOLUTION

The requirements for a license to be approved by the Open Source Initiative as an Open Source license are designed to impose conditions that increase the cost of defecting and provide incentives for cooperation. The annotated version of the definition explains how each of the elements help achieve this goal. See http://www.opensource.org/docs/definition.php

RECIPROCAL AND NON-RECIPROCAL LICENSES

There are two basic types of Open Source licenses that have different incentives and effects. The GNU General public license and other reciprocal type licenses require that anyone who creates a derivative work based on a GPL licensed code must distribute the new work under a license compatible with the GPL that grants the same rights that the developer received with the original code.

A second class of Open Source licenses allow the derivative work to be licensed with a license that is more restrictive than the original code. Most non-reciprocal Open Source licenses allow the Open Source code to be combined with proprietary code to create a new proprietary product.

OS3A FOLLOWS A DUAL LICENSE MODEL

The Open Source SAS® Software Applications project has adopted a dual-license model with four licenses to choose from. Hopefully the developers of each sub-project will choose a single license for each sub-project. The GNU GPL is available as the reciprocal license compatible with most other Open Source code. The Eclipse Public License is the choice for a non-reciprocal license that can be used to encourage use of the code in proprietary business systems. The Lesser General Public License is the preferred license for SAS Macro Language libraries and libraries of formats and ODS templates that might be called by Open Source or proprietary code.

OPEN SOURCE PROCESSES – A FOSS HOW-TO

TECHNOLOGY A PROJECT NEEDS

Fogel (2005) cites a variety of technical infrastructure necessary for an Open Source project. Fortunately the cost to acquire and use these is minimal.

- Web site – a one-way funnel from the project team to the general public.
- Mailing Lists – the primary method for team members to contact one another. Fogel strongly recommends using list management software.
- Version control – as in any typical production shop. Every aspect of the Project is under version control: source code, documentation, web pages, FAQ, and any other crucial project files as appropriate.
- Bug tracking – also functions as a project management central clearing house. Fogel recommends either Concurrent Versions System (CVS) or Subversion (SVN). This also greatly increases transparency to the project.
- Real-time chat for quick and dirty issues among team members as well as the public.

SOCIAL AND POLITICAL INFRASTRUCTURE

Fogel discusses the various forms that successful Open Source projects have employed. One seeming paradox of constant scrutiny and peer review is the desire to remain consistent in order to avoid project “forks”. This is the ability of anyone, at any time, to take (not steal!) all the work to date, copy it to another competing project, and go off in a new direction. This tends to induce a spirit of flexibility and cooperation among project team members, much more so than any authority or monetary directives one might find in a typical company. One popular model is that of the Benevolent Dictator, who has increased authority through common consensus. As a project matures, Fogel indicates that they often evolve into a more democratic model.

MONEY

The involvement of Open Source project members may be strictly voluntary, or it may in fact be their paying day job at a corporation. Many corporations are beginning to appreciate the benefits of long term endorsement of OS projects. Fogel elaborates on several models for financial contribution:

- Sharing the burden – multiple companies can cooperate to create common resources, but with less burden on each individual company.
- Augmenting services – the OS project provides added value to the company’s main selling point.
- Supporting sales of hardware – hardware is useless without quality software to run on it; and high quality well supported software that is free adds a great deal of value.
- Competition with other companies in their sector – this is a natural extension to the point about supporting hardware sales.
- Marketing – increases brand perceived value and visibility.
- Dual licensing – OS projects can be leveraged into traditional proprietary licenses.
- Donations – Fogel recommends careful transparency about how any funds or materials donated to a project are utilized.

COMMUNICATIONS

Fogel makes a strong case that communication skills trump advanced technical skills in an Open Source project. He cites the case of a valuable contributor who, when pressed, could not obtain corporate copyright permissions because he was thirteen years old! High quality communication speaks volumes and produces great leverage in an OS framework. This involves paying close attention to writing style, proper spelling and grammar, and especially attending to the emotional tone.

PACKAGING, RELEASING, AND DAILY DEVELOPMENT

There are many differences between an Open Source project and the typical corporate structure, which is usually oriented to the master schedule (often at the expense of such minor matters as software quality or work-life balance). Open Source projects tend to have multiple threads running in parallel, not the monolithic approach many corporate teams employ.

While project members are always working with the latest (bleeding edge) version of the software, Open Source projects follow the same major/minor version release cycle of most software development. OS projects usually evolve into the practice of producing *release branches*:

> “The solution to these problems is to always use a release branch. A release branch is just a branch in the version control system (see branch), on which the code destined for this release can be isolated from mainline development. The concept of release branches is certainly not original to free software; many commercial development organizations use them too. However, in commercial environments, release branches are sometimes considered a luxury—a kind of formal ‘best practice’ that can, in the heat of a major deadline, be dispensed with while everyone on the team scrambles to stabilize the main tree.” (Fogel, p. 122)

A nice discussion of how version control software enables this behavior is provided. There are also political aspects to the release schedule, which relates back to the discussions on Infrastructure and Communications above.

MANAGING VOLUNTEERS

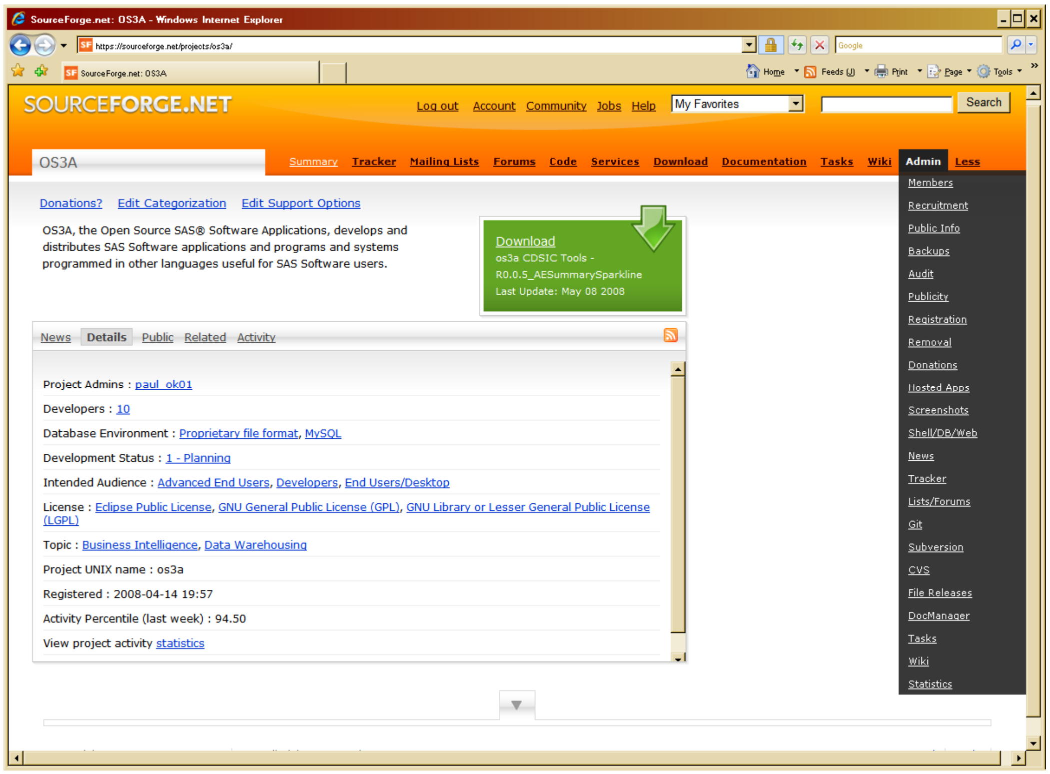
Fogel provides a fascinating discussion on what *really* motivates programmers, what sort of tone one should adopt when requesting project members to perform work, and how to express both praise and criticism. While his comments are intended to apply to the realm of Open Source projects, managers of programmers in any venue would do well to peruse this chapter closely.

LICENSES, COPYRIGHTS, AND PATENTS

Fogel finishes up with a discussion about legal issues as they apply to Open Source licensing. He makes the distinction between the terms *free software*, *Open Source*, *FOSS*, *F/OSS*, and *FLOSS*. He also discusses the vital area of copyright assignment and code ownership – the team would be well served thinking of these issues sooner rather than later.

ON-LINE RESOURCES FOR OPEN SOURCE PROJECTS

THE SOURCEFORGE.NET SITE FOR THE OS3A PROJECT



OS3A, the Open Source SAS® Software Applications, develops and distributes SAS Software applications and programs and systems programmed in other languages useful for SAS Software users.

Download
os3a CDSIC Tools - R0.0.5_AESummarySparkline
Last Update: May 08 2008

Project Admins : paul_ok01

Developers : 10

Database Environment : Proprietary file format, MySQL

Development Status : 1 - Planning

Intended Audience : Advanced End Users, Developers, End Users/Desktop

License : Eclipse Public License, GNU General Public License (GPL), GNU Library or Lesser General Public License (LGPL)

Topic : Business Intelligence, Data Warehousing

Project UNIX name : os3a

Registered : 2008-04-14 19:57

Activity Percentile (last week) : 94.50

View project activity statistics

Shown above is main page of the Open Source SAS® Software Applications project on SourceForge.net. Each of the areas shown across the page at the bottom of the orange section has a menu with several options like the Admin menu shown on the right.

The site offers all of the technology needed for a project recommended by Fogel as described above. Within this main site, several quasi-independent sub-projects could be developed. Some of the sub-projects could be common libraries that are used by several applications that are specific to a particular domain.

FUTURE OF THE OS3A PROGRAM – WHAT IS TO BE DONE?

WHAT ARE SOME 'COOL CODE' PROJECTS TO PURSUE?

OS SAS® APPs

One of the observed success factors for Open Source projects is an initial objective that is attractive to developers so that they will come forward and participate. It seems that it will take the brainstorming and discussion of many potential participants to come up with these ideas. Here are a few ideas from the authors to get us started:

Code Documentation/Organization Some have suggested that instead of a central library or repository of Open Source SAS code what is needed is a way to locate all the SAS code that is already available on the Internet. There are a number of systems for facilitating the creation of external documentation from information within program files that have been developed over the last few years. These are: CodeDoc by Jeff Wright and his colleagues at Thotwave, LLC., DocItOut by Choon-Chern Lim, and SaviDoc by Alan Churchill. Each of these can be used to generate XML or HTML program documentation that could be stored on the Internet or an organization's intranet. A search engine could then be used to find a program with the desired functionality.

Metadata Application Programming This is one of a class of projects that demonstrates innovative programming techniques. Data values are removed from code and stored in tables that are accessed when the data is needed in the program. When this technique is adopted, the need for a supporting set of management tools becomes apparent.

Code Design Standards Some of the co-authors on this paper are also proposing an effort among Pharmaceutical companies to develop SAS programming standards for the industry. Teams of programmers could also develop their own programming standards and use the OS3A project to demonstrate how these standards and good programming methods can work together for SAS programmers. The OS3A project will also give SAS programmers the opportunity to learn good practices in unit and acceptance testing and skill in the use of standard code version control systems.

ODS Templates and Methods ODS will continue to develop into a key resource for SAS users. Sophisticated templates can be developed and shared with the SAS user community. New methods can be demonstrated with practical programming.

SAS/IML Studio Applications The new possibilities for implementing innovative new statistical methods and reporting needs a library of examples. How can SAS analysis be combined with the functionality of other products?

Clinical Reporting A companion paper[1] by the authors proposes an Open Source effort to develop a clinical trials reporting system using SAS programming to save cost and effort in the Pharmaceutical industry.

CONCLUSSION

Open Source development offers a new opportunity for SAS programmers to work together on projects to benefit the entire community of SAS users. Developers can sharpen their skills using programming best practices and the exchange of ideas with their peers. SAS users will choose to do what interests them instead of just what pays their livehood.

REFERENCES

Benkler, Yochai. 2006. **The Wealth of Networks: How Social Production Transforms Markets and Freedom**. New Haven, CT: Yale University Press, http://cyber.law.harvard.edu/wealth_of_networks/Main_Page, (Feb. 21, '09)

Deek, Fadi P., and James A. M. McHugh. 2008. **Open Source Technology and Policy**. New York, NY: Cambridge University Press

DiBona, Chris, Sam Ockman, and Mark Stone, Eds. 1999. **Open Sources: Voices from the Open Source Revolution**. Sebastopol, CA: O'Reilly & Associates, Inc.

DiBona, Chris, Danese Cooper, and Mark Stone, Eds. 2006. **Open Sources 2.0: The Continuing Evolution**. Sebastopol, CA: O'Reilly & Associates, Inc.

Feller, Joseph, Brian Fitzgerald, Scott A Hissam, and Karim R. Lakhani. 2005. **Perspectives on Free and Open Source Software**. Cambridge, MA: The MIT Press.

Fogel, Karl. 2006. **Producing Open Source Software: how to run a successful free software project**. Sebastopol, CA: O'Reilly & Associates, Inc. http://producingoss.com/, (Feb. 21, '09)

Lindberg, Van. 2008. **Intellectual Property and Open Source**. Sebastopol, CA: O'Reilly Media, Inc.

Pavlicek, Russell C. 2000. **Embracing Insanity: Open Source Software Development**. Indianapolis, IN. Sams Publishing.

Raymond, Eric S. 1999. **The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary**. Sebastopol, CA: O'Reilly & Associates, Inc.

Rosen, Lawrence. 2004. **Open Source Licensing: Software Freedom and Intellectual Property Law**. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference, Inc.

St. Laurent, Andrew M. 2004. **Understanding Open Source & Free Software Licensing**. Sebastopol, CA: O'Reilly & Associates, Inc.

Tapscott, Don, and Anthony D. Williams. 2006. **Wikinomics: how mass collaboration changes everything**. London: Penguin Group, Ltd.

Williams, Sam. 2002. **Free as in Freedom: Richard Stallman's Crusade for Free Software**. Sebastopol, CA: O'Reilly & Associates, Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Paul OldenKamp
Clinical Trials Systems Analyst/Prog.
Seattle Children's Res. Inst.
8321 Twenty-ninth Avenue NW
Seattle, WA 98117
Tel: 206.884.7539
paul.oldenkamp@seattlechildrens.org

OS3A Project Administrator
Open Source SAS® Software Applications.
https://sourceforge.net/projects/os3a/
OS3A email:
https://sourceforge.net/sendmessage.php?touser=2044988

Paul D. Hamilton
Biostatistical Programming Sr. Mgr.
Amgen Inc.
1201 Amgen Court West
Seattle, WA 98119
Tel: 206.265.7689
hamiltop@amgen.com

Ann Marie Martin
Director, Global Statistical Programming
UCB Pharma s.a.
Chemin du Foriest
B-1420 Braine l'Alleud
Belgium
Tel: +32-2-386 35 57
Cell: +32-494-578 315
annmarie.martin@ucb.com

Dante DiTommaso
Statistical Applications Team Leader
F. Hoffmann-La Roche AG
Methodology and Innovation
Malzgasse 30 (B. 670 / R. 308)
CH-4052, Basel, Switzerland
Tel: +41 61 68 74314
dante.di_tommaso@roche.com