

JOURNAL OF INFORMATION SYSTEMS APPLIED RESEARCH

In this issue:

4. **Creating an Audio Conferencing Application on Android Smart Phones**
Jui Sun, University of North Carolina Wilmington
Ron Vetter, University of North Carolina Wilmington
Bryan Reinicke, University of North Carolina Wilmington

22. **Open Source Software Volunteerism vs. Motivating Potential of Primary Employment: Suggestions for a Research Agenda**
Donald A. Carpenter, Colorado Mesa University

31. **Rocky Relationships: Enterprise Resource Planning and Supply Chain Management**
Jack Crumbly, Tuskegee University
Meg Fryling, Siena College

40. **The Impact of Intra-Organizational Social Networking Sites on Impression Formation**
Jeff Cummings, University of North Carolina Wilmington

The **Journal of Information Systems Applied Research** (JISAR) is a double-blind peer-reviewed academic journal published by **EDSIG**, the Education Special Interest Group of AITP, the Association of Information Technology Professionals (Chicago, Illinois). Publishing frequency is currently quarterly. The first date of publication is December 1, 2008.

JISAR is published online (<http://jisar.org>) in connection with CONISAR, the Conference on Information Systems Applied Research, which is also double-blind peer reviewed. Our sister publication, the Proceedings of CONISAR, features all papers, panels, workshops, and presentations from the conference. (<http://conisar.org>)

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the JISAR journal. Currently the target acceptance rate for the journal is about 45%.

Questions should be addressed to the editor at editor@jisar.org or the publisher at publisher@jisar.org.

2013 AITP Education Special Interest Group (EDSIG) Board of Directors

Wendy Ceccucci
Quinnipiac University
President - 2013

Leslie J. Waguespack Jr
Bentley University
Vice President

Alan Peslak
Penn State University
President 2011-2012

Jeffry Babb
West Texas A&M
Membership

Michael Smith
Georgia Institute of Technology
Secretary

George Nezlek
Treasurer

Eric Bremier
Siena College
Director

Nita Brooks
Middle Tennessee State Univ
Director

Scott Hunsinger
Appalachian State University
Membership Director

Muhammed Miah
Southern Univ New Orleans
Director

Peter Wu
Robert Morris University
Director

S. E. Kruck
James Madison University
JISE Editor

Nita Adams
State of Illinois (retired)
FITE Liaison

Copyright © 2013 by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to Scott Hunsinger, Editor, editor@jisar.org.

JOURNAL OF INFORMATION SYSTEMS APPLIED RESEARCH

Editors

Scott Hunsinger
Senior Editor
Appalachian State University

Thomas Janicki
Publisher
University of North Carolina Wilmington

JISAR Editorial Board

Samuel Abraham
Siena Heights University

Doncho Petkov
Eastern Connecticut State University

Jeffrey Babb
West Texas A&M University

Samuel Sambasivam
Azusa Pacific University

Wendy Ceccucci
Quinnipiac University

Bruce Saulnier
Quinnipiac University

Ken Corley
Appalachian State University

Mark Segall
Metropolitan State University of Denver

Gerald DeHondt II

Anthony Serapiglia
St. Vincent College

Mark Jones
Lock Haven University

Li-Jen Shannon
Sam Houston State University

Melinda Korzaan
Middle Tennessee State University

Michael Smith
Georgia Institute of Technology

James Lawler
Pace University

Karthikeyan Umapathy
University of North Florida

Terri Lenox
Westminster College

Stuart Varden
Pace University

Michelle Louch
Robert Morris University

Leslie Waguespack
Bentley University

Cynthia Martincic
St. Vincent College

Laurie Werner
Miami University

Fortune Mhlanga
Lipscomb University

Bruce White
Quinnipiac University

Muhammed Miah
Southern University at New Orleans

Peter Y. Wu
Robert Morris University

George Nezelek

Ulku Yaylacicegi
University of North Carolina Wilmington

Alan Peslak
Penn State University

Open Source Software Volunteerism vs. Motivating Potential of Primary Employment: Suggestions for a Research Agenda

Donald A. Carpenter
dcarpent@coloradomesa.edu
Colorado Mesa University
Grand Junction, CO 81506 USA

Abstract

The open source software phenomenon can be studied from a variety of perspectives. Given that much of the work on open source project has been accomplished by volunteers, one popular thread has been to examine the motivation of those who engage in open source software development. If motivation wanes, predictably there would be fewer contributions to open source projects. This paper reviews relevant literature on open source software, highlighting a recently published study that offers a prescription for future research based on social practice theory. The literature review herein also examines research on the motivation of computer programmers, the motivation of volunteers, and the personality of programmers. Next this paper weaves together those various research threads. This paper concludes with recommendations for a different direction for expanding the research on open source software developers' motivation by blending the extensive findings on job design of computer programming positions, research on motivation of volunteers, and research on programmer personalities. The paper ends with recommendations for future research hypotheses.

Keywords: open source software, programmer, motivation, work design, volunteerism

1. INTRODUCTION

The open source software (OSS) movement has grown over three decades—longer if free software (FS) is included—to the point where OSS resulted in \$60 billion annual consumer savings in 2008 (OSS, 2012). Nearly 32,000 developers had contributed over five billion bytes of OSS code by 2002, with 74% of the code contributed by 10% of the developers (OSS, 2012). The advent of the Open Source Initiative (OSI) in 1998 formalized structure of the OSS community and quality of the software produced through standards and licensing (OSI, 2012). A few years ago, it was assumed that most of the OSS code is contributed by volunteers on their own time (OSS, 2012).

In the past few years, the situation has changed. A 2004 InformationWeek survey “found that

67% of companies use OSS/FS products, with another 16% expecting to use OSS/FS in 2005” (Wheeler, 2012). That has caught the attention of companies such as Google, Red Hat, IBM, Intel, Apple, Oracle, and Sony, all of which have released their own OSS products.

Such an economically important activity certainly warrants examination from multiple perspectives. One such recent angle has been to examine the motivation of the programmers who collaborate to create OSS primarily on their own time and without remuneration, i.e., as volunteers. At one time those accounted for the majority of OSS code. However, the playing field has changed. There are those instances of major corporations (as listed in previous paragraph) paying programmers to write OSS code. Another example relates to the popularity of Linux as the primary operating system for

servers on the Internet: 75% of the code of Linux is contributed by paid programmers. There remains a substantial portion of the total OSS developers who contribute to OSS voluntarily (Wheeler, 2007).

Many scholars have written on the topic of motivation of OSS contributors. Many of those seem to consider OSS contribution as a completely or predominately voluntary activity. However, no instances were found of any examination similar to that done herein, which addressed both paid and voluntary contributions to OSS.

This paper examines the literature on motivation of OSS developers, on motivation of programmers in general, and on motivation of volunteers. Given that motivation is individualized, this paper also examines the literature on research into the personalities of computer programmers. Then, it discusses the implications of the findings in the literature regarding directions for future research. It draws conclusions that previously recommended research paths might not get at the question thoroughly. This paper then suggests other opportunities for future research to get to the bottom of the question "What really motivates volunteer OSS developers?"

2. LITERATURE REVIEW

Motivation to Create OSS (OSS web sites)

Before turning to the scholarly literature, some enlightenment can be gained from considering what OSS developers state as their motivation to engage in OSS development. OSS developers' opinions are more likely to be found on blogs or on their own web sites rather than in the scholarly literature. Four sample web sites pertaining to OSS development provide the insights presented below. It should be noted that "motivation" in lay terms may be used to mean extrinsic "goals," "aims," or even "philosophies," rather than intrinsic "psychological motivators" used in scholarly research (Luthans, 2011, p. 157).

Hughes (2004) emphasizes that "The open source software I have seen is too good to be done by people who aren't very talented, experienced engineers" and that many "are in fact people who have a regular full time job." He emphasizes that there exists a gatekeeper on each OSS project to enforce quality, plus quality

is enforced by standards and licensing rules. Hughes also explains the following as "a few of the better known reasons" why talented, employed programmers moonlight in OSS, conditions not necessarily present in day jobs.

- "plaudits from peers,"
- "technological challenges,"
- opportunity to be one's own boss,
- sharing common interests with others,
- making an important contribution to something that will endure over time,
- building additional work experiences.

Rhodes (2008) takes a different perspective, noting that "Most open source software projects were created by a programmer who needed a piece of software to accomplish a certain task. ... the programmer decided to share it with the world by publishing it under an open source license." Subsequent programmers discover that software and make modifications to fit their specific needs, giving rise to open source communities and even to open source businesses.

A third explanation blends elements of Hughes and of Rhodes. Open Advocacy (2012) emphasizes that "By sharing the source, people are able to create a better product working together than they could working apart." That sharing builds camaraderie desired by the participants, encouraging a common opinion that all software source code should be shared, i.e., open source.

Wheeler (2007) reports a 2002 survey found that the top OSS developer motivations were

- intellectually stimulating (44.9%)
- improves skill (41.3%)
- work functionality (33.8%)
- code should be open (33.1%)
- non-work functionality (29.7%)
- obligation from use (28.5%)

from which he derived four groups:

- Believers (19%): believe source code should be open,
- Learning and Fun (29%): for non-work needs and intellectual stimulation,
- Hobbyists (27%): need the code for a non-work reason,
- Professionals (25%): for work needs and professional status.

Motivation to Create OSS (scholarly)

This paper now turns to the scholarly literature on the motivation to write OSS. Many scholars have published on the topic. Von Krogh, et al. (2012) have done a yeoman's job compiling the results of dozens of those OSS studies. Hence, there is no need to duplicate their efforts herein. Rather, highlights of their findings are in order.

Von Krogh, et al. explain that of the dozens of research frameworks in the scholarly works they examined, "the most frequent framework by far has been the distinction between intrinsic and extrinsic motivation in self-determination theory (SDT)," a mainstay psychological and organization behavior theory posited by Deci & Ryan (1985). Subsequently, von Krogh, et al. arbitrarily group the outcomes of forty research studies they examined into ten motivational categories under three headings, as illustrated in by following list.

- Intrinsic motivators
 - Ideology
 - Altruism
 - Kinship
 - Fun
- Internalized Extrinsic
 - Reputation
 - Reciprocity
 - Learning
 - Own-use
- Extrinsic
 - Career
 - Pay (p. 654)

Von Krogh, et al. state that "existing literature does not provide satisfactory answers to three differentiated questions as to why this [OSS] phenomenon exists." (p. 650). Those three questions are framed in the context of the values of social practice theory advocated by philosopher Alasdair MacIntyre (1981). Each question is well supported by theory and research propositions. The three questions, which are the basis for further research advocated by von Krogh, et al. (pp. 666-8) are:

- "How and why do OSS developers produce high-quality software (goods) when they do?"
- "Why do OSS developers change institutions?"
- "Why do developers sustain the social practice of OSS development?"

In Von Krogh, et al.'s concluding remarks is: "there is ample room to investigate motivation from a multitude of perspectives and methodological approaches." (p. 671)

A 2002 survey (Knosh, et al.) of 2,784 open source and free software developers indicated a median developer age of 26 years and a median starting age of those developers of 22 years (p. 8). Nearly 70% of the contributors spend ten or fewer hours per week on such work. The following list presents reasons (and respective percent of responses) to join an OSS community and to stay in an OSS community (p. 45).

- Make money (4.4, 12.3)
- Software should be open (30.1, 37.9)
- Limit power of big software firms (19, 28.9)
- Solve a specific problem (29.7, 29.6)
- Get help w/good software product (23.8, 27)
- Distribute software products (8.9, 10)
- Build self reputation (9.1, 12)
- Improve others' sw products (33.7, 39.8)
- Improve job opportunities (23.9, 29.8)
- Participate in OSS/FS scene (30.6, 35.5)
- Share knowledge and skills (49.8, 67.2)
- Learn and develop new skills (78.9, 70.5)
- Participate in cooperative form (34.5, 37.2)

Motivation of Computer Programmers

While Van Krogh, et al. present a literature review that supports their research agenda, they miss a large portion of research on motivation of computer programmers. Hackman & Oldham (1980) applied their job characteristics model (JCM) and job diagnostic survey (JDS) to 6930 workers in 56 organizations and in dozens of jobs in nine general job categories (p. 316). Their findings were reported as Growth Need Strengths (GNS) of the workers and Motivating Potential Score (MPS) of jobs. The study has been replicated many times in many settings with similar results each time. JCM concepts are components of other more complex, more individualized motivational theories such as Deci and Ryan (Carpenter, 2003).

JCM research indicates that high intrinsic work motivation (as well as high quality work performance, high work satisfaction, low turnover, and low absenteeism) result from three critical psychological states: (1) experienced meaningfulness of work, (2) experienced responsibility for outcomes of work, and (3) knowledge of the results of activities. Those three critical psychological states result from five

core job dimensions: (1) skill variety, (2) task identity, (3) task significance, (4) autonomy, and (5) feedback. There is no inherently good or bad level of MPS or GNS. Rather, it is good when an employee's GNS matches the MPS of his job and it is bad when there's a mismatch (Hackman & Oldham, p. 90).

Cougar & Zawacki (1980) expanded Hackman & Oldham's research to include computer professionals (i.e. programmers and systems analysts) and computer operators. Their original study and many subsequent studies demonstrate that programmers/analysts have significantly higher GNS than the next highest category (i.e., other professionals) but their jobs have nearly identical MPS as other professionals' jobs. (p. 21)

An interesting subset of Cougar & Zawacki's findings pertains to the component of satisfaction levels of programmers. While programmers have a higher level of general satisfaction than do other professionals, their levels of satisfaction with their co-workers and with their supervisors tends to be significantly less than those of other professionals. (p. 17)

Carpenter, et al. (2004) administered the JDS to 43 information technology (IT) employees in 13 different IT job categories. Only one programmer and one analyst were included in the sample. There was no statistically significant difference between MPS of the subjects' jobs and that of computer professionals in Cougar & Zawacki. However, subjects' GNS was statistical equivalent to GNS of general white collar workers, which is below that of general professionals, which in turn is below that of computer professionals. This indicates a greater mismatch for IT jobs in general than there is for analysts and programmers, between their motivational needs and the motivation that their jobs provide.

Motivation of Volunteers

Given that the majority of OSS is developed via a volunteerism model, it is important to look at the literature on motivating volunteers. Bang & Ross (2009, p. 61) studied responses from 254 volunteers at sporting events using a specialized instrument for measuring motivation of sporting event volunteers. Their findings indicate sporting event volunteers' motivations are grouped into seven factors: (1) expression of values, (2) community involvement, (3)

interpersonal contact, (4) career orientation, (5) personal growth, (6) extrinsic rewards, and (7) love of sports.

Millette & Gagné (2008) analyze the motivation of 230 volunteers in a community clinic by applying Hackman & Oldham's JCM model. Among other findings, they note that MPS of volunteer positions is positively correlated with workers' levels of satisfaction (p. 17). That indicates that the JCM concepts of motivation apply similarly to volunteers as they do to paid workers and that the JDS is usable in volunteer settings.

Broekmeier (2010) warns against romanticizing the volunteer aspect of free and open source software (F/OSS or FLOSS), suggesting it is a norm that contributions begin during college when the volunteers have time and falls off if the graduate does not get a job writing FLOSS as a result of his contributions. He also seems to doubt whether volunteers are any differently motivated than paid contributors.

Berdou (2006) suggests that paid FLOSS contributors drive the major free and open source (F/OS) projects, while volunteers operate on the periphery. She notes that the paid FLOSS contributions have "been largely overlooked in the F/OS literature." (p. 201) The notion that motivations of paid versus volunteer programmers differ "has been challenged on several fronts by researchers who draw attention to the interconnections between the two spheres of economic activity." (p. 202)

Personality of Programmers

The Myers-Briggs Type Indicator (MBTI) has been applied in hundreds of studies. By means of a subject-answered survey, MBTI categorizes a subject's personality into one of sixteen types that are measured on four dimensions, as indicated by the pairs of terms in the list below. Studies show occurrences in the United States adult population to be distributed as follows, per CAPT (2012):

- Extroversion (E) 45-53%
- Introversion (I) 47-55%

- Sensing (S) 66-74%
- Intuiting (N) 26-34%

- Thinking (T) 40-50%
- Feeling (F) 50-60%

- Judging (J) 54-60%
- Perceiving (P) 40-46%

It is important to note that these ranges are probably different in other cultures, as some traits are valued and nurtured differently in various cultures. It is also important to recognize that people often behave differently than would be predicted by their personality types. (E&I, 2012)

Because they will come in handy later, the author inserts at this point in the literature review, four definitions. "Introversion" is "the act of directing one's attention toward or getting gratification from one's own interests, thoughts, and feelings" (Introversion, 2012). "Sensing" is "paying attention to physical reality" rather than intuition (Sensing, 2012). "Thinking" is "the action of using one's mind to produce thoughts" (Thinking, 2012). "Judging" is "forming an opinion about through careful weighing of evidence and testing of premises (Judging, 2012). The other four MBTI anchor terms can be derived as functional opposites of the ones just presented.

Many web sites provide information as to which jobs are best suited to which personality types. One such source (Career Matches, 2012) indicates that "computer programmer" is a good match for four of the eight personality types that include introversion but for only one of the eight personality types that include extroversion. Also, computer programmer is a good match for five of the eight personality types that include thinking, but none that include feeling; three that include judging but two that include perceiving; three that include intuiting, but two that include sensing. It is unknown as to whether those listings are supported by data and statistical analysis. Therefore, it would be irresponsible to conclude from that rundown that computer programming aligns very strongly with thinking and strongly with introversion. Nonetheless, it is an interesting set of information.

Similarly, one should be cautious about statements not supported by sound research. For example, in one essay is found: "I also observed that most really good software engineers were ENTJ, INTJ, ESTJ, ISTJ, ISFJ, and ENTP (Hardiman 1997, p. 10). Questionable also is an approach such listing software programmer job requirements, mapping those to a list of soft skills, and in turn arbitrarily mapping those to the eight anchors on the four MBTI scales in

order to determine that "Most programmers are introvert (I), sensing (S), thinking (T) types." (Capretz & Ahmed, 2010, p.10) While the latter holds merit as a hypothesis, there is no quantitative data in that paper to support the claim.

Bentley (2005) cites McConnell (1999) as estimating that 20-40% of programmers are ISTJ or INTJ and that 80-90% of programmers have personalities that include the thinking element, but that programmers are evenly split between sensing and intuiting. That would imply that thinking and intuiting are overrepresented in programmers by comparison to the general U.S. population, according to the CAPT data on the previous page, above. Bentley himself concludes that the three great virtues of programmers are laziness, impatience, and hubris (2005, p. 7). However, per Bentley's definition, "efficient" and "quality-oriented" might fit equally well.

In Capretz (2003), the literature review discusses the results of four studies that used MBTI to measure programmer personality types. Capretz summarizes: "The common thread running through the results of these studies in the prevalence of introverts, thinking, judging, and almost as many sensing as intuitive among software professionals." (p. 209) When this author examines the data in those four studies in light of the CAPT ranges, thinking and intuiting are overrepresented in the programmers studied. On the other hand, introversion and judging percentages appear to be within CAPT's ranges.

Capretz's own study of 100 software engineers revealed 57% I, 67% S, 81% T, and 58% J (p. 211). By visual comparison to the CAPT ranges for the general U.S. population, introversion is slightly over-represented and thinking is hugely over-represented, while sensing and judging are within the CAPT ranges, although Capretz states otherwise. Capretz notes that 24% of his sample was scored as ISTJ, compared to 11.6% of the general U.S. adult population (p. 212).

In addition to reporting on GNS of programmers and MPS of their jobs, Cougar & Zawacki (1980) also reported on a personality aspect of programmers and system analysts, specifically their social needs strengths (SNS). Subsequent research with other programmers in other settings confirmed their findings. SNS is defined as "an individual's need to interact with others" (p. 23). Programmers and analysts have a substantially and statistically lower SNS than all

other professionals. It would be a logical leap to connect low SNS with introversion. However, the research did not explore that connection.

In addition to considering personality types, some researchers consider two-letter combinations of the MBTI variables as definitions of "temperaments." Keirsey & Bates (1978) present four temperaments: SP, SJ, NT, and NF, which they label as artisans, guardians, rationals, and idealists, respectively. Temperament letter combinations can be identified in the findings of several of the research studied discussed above. However, Keirsey (2012) does not give potential employment connections for the temperaments. Rather, it specifies programming or related positions as possibilities for a few of the MBTI types in guardian and rational groups. Here also, it would be dangerous to assume. Yet, in other studies, the SJ and NT letters are associated with programmers.

An aspect related to personality is behavior, the difference being that behavior is observable whereas personality is internal (Merrill & Reed, 1981, p. 7). Patterns of behavior can be grouped by four social styles (p. 2). While there is no stated correlation between social styles and preferred employment type, the thinking orientation of the analytical style (TRACOM, 2012) aligns with the overrepresentation of the thinking (T) variable in some of the above stated research. It is this author's observation over the years that most programmers also tend to exhibit most of the other characteristics of the analytical style.

3. DISCUSSION

Pursuing questions of OSS programmer motivation by applying the concepts of social practice theory (as advocated by von Krogh, et al.) might indeed yield significant results. Replication of that research with multiple populations in multiple settings will be a long term endeavor. It will be interesting to follow the process through the scholarly publications that result.

One challenge with treating the OSS community from a sociological perspective is the rapid rate of change occurring in that community. It is no longer the predominately-volunteer set of activities that it was a few short years ago. Major corporate players are shaping the game as much if not more so than are volunteers.

Moreover, some of recommended research points about quality suggested by von Krogh, et al. might be answered by simply looking to the standards set by the Open Source Initiative. Other suggested research points about how developers shape organizations might be answered by examining the reported recent heavy involvement of large corporations which might be the real shapers.

This author thinks important answers about volunteer OSS developers' motivation lie elsewhere. Specifically, one can readily build upon a large body of research on motivation and personalities of programmers. The reason for drawing on literature related to personality is that motivation is individualized. Personality-oriented research gives clues to the individualistic nature of motivation. OSS programmers are a subset of all programmers. Measurement of OSS developers' GNS, SNS, MBTI types, and social styles, and MPS of their jobs would provide linkages back to extensive prior research.

If MPS was measured for both the programmers' paid jobs and for their volunteer OSS activities, differences might be seen that explain that they engage in OSS development because doing so provides important elements that are lacking in their paid jobs. Such measurement could also be tied to the literature on volunteers as it has been shown that JDS can be appropriately applied to volunteer settings.

In particular, conducting such research could show OSS involvement is the preferred form of volunteerism for certain MBTI types who have the necessary skills. Those with lower SNS might find creating OSS to be a more desirable outlet than volunteer work that requires more face-to-face social interaction. The same could be said for the thinking or any other personality component in regard to OSS development. Perhaps programmers volunteer in OSS communities because they are stimulated by the additional thinking required of them, or the additional sensing, or the additional judging. Perhaps they do not get enough of those in their paid jobs.

Ultimately, the analysis could point out what needs to be improved in programmers' paid jobs. That would allow the application of job design to improve those jobs' motivating potential. That is not to say that employers would improve jobs solely in order to reduce the amount of time their

employees spend moonlighting on OSS. Rather, the goal would be to improve productivity, satisfaction, absenteeism, and turnover for those paid jobs. Employers should no more object to their employees' volunteer involvement in OSS than to other employees volunteering for other community projects.

Despite the reality that motivation is highly individualized, averages can be established to explain groups of typical subjects. As those analyses are performed, there are two groupings that should be established. As noted above, a growing portion of OSS is built as part of programmers' job assignments. The data for those should be segregated from the data for those who engage in OSS development voluntarily. A third category would be those who create OSS both as part of a paid job and as a volunteer.

Another demographic breakdown to consider is based on the off-used 80/20 rule of work. Above it is reported that 10% of the OSS developers have done 74% of the work. Perhaps those most heavily involved in developing OSS do not have paid jobs. It is unknown as to whether that ratio still holds. If it does hold, and if those in that minority are paid workers, perhaps they are less motivated by their paid jobs than are those in the majority. Or perhaps they are doing the OSS work because it is part of their paid jobs.

Other worker, job, and employer demographics should be collected as well as data about the type of OSS and the amount of time expended. It would be interesting to learn if there are age, gender, education-level, or other considerations that correlate with involvement in OSS. Given today's prevalence of offshore outsourcing of IT jobs, including programming, one imperative of such research should be to measure whether programmer and job characteristics as well as OSS involvement vary by geographic location.

One set of data that would greatly interest this author is regarding those programmers who are not involved in OSS development activities. Some say that OSS engagement is the new normal among programmers. If so, it might be more revealing to learn why some (most?) programmers do not become involved in OSS creation. Ultimately, that could prove to be more revealing than answers to why programmers do engage in OSS development.

4. CONCLUSIONS

This paper was written partially in response to a recent publication (Von Krogh, et. al, 2012) that recommends a social practice theory-based research agenda to determine why programmers engage in OSS development. This author thinks that it would be more productive to tie the OSS question to the large body of prior research on programmer motivation, programmer personality, and volunteer motivation. Those instruments are readily available. Hence, those results could be generated more quickly, which is important in a rapidly changing environment as the OSS movement. Those findings might be more widely received as more scholars have been exposed to those concepts than to social practice theory.

This author anticipates such research to yield several specific results that align with the notion that OSS involvement is the norm for younger programmer and that the OSS movement now is driven by corporations. They could be used as hypotheses by those who might engage in that research. It is expected such research will show:

- Extensive involvement in OSS development will vary with certain demographics, especially age, education level, time on the job, time proximity from college graduation, and geographic location.
- Demographics of OSS developers will show that a majority of OSS contributions are now being made by paid developers.
- By comparison to those who are not engaged or only tangentially engaged, those programmers who are engaged extensively in OSS development will have marginally higher growth need strengths and different personality types and social styles. However those measures of all OSS contributors will not be much different than all programmers.
- The paid jobs of those who generate OSS on their own time are deficient in terms of their motivating potential, thereby falling short of the employees' motivational needs. However, those jobs are no less deficient than the jobs of programmers in general.
- Outside OSS engagement fills some motivational void in those insufficient paid jobs.

- Those who spend extensive amount of time engaged in OSS development will have personality types, social styles, and social needs strengths that are substantially different from the general programmer population.
- OSS development allows programmers who possess certain personality types to engage in volunteerism that is not people-oriented.
- Developers' goals for their engagement in OSS will vary depending on whether they are volunteers or paid contributors.

The caveat to consider as such a research project begins is to pay attention to the rapid change in the OSS community. Both the above set of opinions and the hypotheses that result are subject to change with change in the nature of the OSS phenomenon. A thorough review of then-current literature generated by the OSS community might serve the researchers well as they deal with those ongoing inevitable changes.

5. REFERENCES

- Bang, H., & Ross, S. D. (2009). Volunteer motivation and satisfaction. *Journal of Venue & Event Management*, 1(2). 61-77.
- Bentley, J. E. (2005). Laziness, impatience, hubris: personality traits of a great programmer. Proceedings of Southeast SAS Users Group (SESUG). Retrieved July 5, 2012 from http://analytics.ncsu.edu/sesug/2005/SERO^_05.pdf.
- Berdou, E., 2006, Insiders and outsiders: paid contributors and the dynamics of cooperation in community led F/OS projects. In Damiani, E., Fitzgerald, B., Scacchi, W., Scotto, M., & Succi, G., Eds. *International Federation for Information Processing, 203, Open Source Systems*. Boston: Springer. 201-208.
- Brockmeier, J. (2010) Motivation & contributions in open source: stop romanticizing unpaid contributions. Ostatic Blog. Retrieved July 10, 2012 from <http://ostatic.com/blog/motivation-and-contributions-in-open-source-stop-romanticizing-unpaid-contributions>.
- Carpenter, D.A. (2003). Connecting Logotherapy with motivational theories: Plan for research & consulting regarding meaningfulness. Proceedings of *World Congress Logotherapy*.
- Carpenter, D. A., Slauson, G.J., Snyder, J., & Winniford, M.A. (2007). Motivating & managing computer personnel in 21st century: expanding pilot study. *Proceedings of Mountain Plains Management Conference*. 270-279.
- Capretz, L. F. (2003) Personality types in software engineering. *International Journal of Human-Computer Studies*, 58(2). 207-214.
- Capretz, L.F., & Ahmed, F. (2010). Making sense of software development and personality types. *IT Professional*, 12(1). 6-13.
- Career matches by personality type. (2012). Retrieved July 5, 2012 from <http://www.my-personality.info/personality-types/careers/>.
- Center for Applications of Psychological Types (CAPT). (2012). Estimated frequencies of the types in the United States population. Retrieved Jul 7, 2012 from <http://www.capt.org/mbti-assessment/estimated-requencies.htm>.
- Couger, J. D., & Zawacki, R. A. (1980). *Motivating and Managing Computer Personnel*. New York, John Wiley and Sons.
- Deci, E. L., & Ryan, R. M. (1985). *Intrinsic Motivation and Self-Determination in Human Behavior*. New York: Plenum Press.
- Extraversion and introversion (E&I). (2012). In *Wikipedia, The Free Encyclopedia*. Retrieved on July 8, 2012, from http://en.wikipedia.org/w/index.php?title=Extraversion_and_introversion&oldid=500919123.
- Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. (2002). Free/libre & open source software: survey and study, Part 4. International Institute of Infonomics, University of Maastricht, Netherlands.
- Hackman, J. R., & Oldham, G. R. (1980). *Work Redesign*. Reading, MA: Addison-Wesley.
- Hardiman, L. T. (1997). Personality types and software engineers. *Computer*, 30(10). 10.
- Open Source Initiative (OSI). (2012). History of Open Source Initiative, Retrieved Jul 9, 2012 from <http://www.opensource.org/>.

- Hughes, J. (2004). Why do people write open source software? *OpenXtra*. Retrieved June 14, 2012 from <http://www.openxtra.co.uk/articles/why-do-people-write-open-source-software>.
- Introversion. (2012). In Merriam-Webster online Dictionary. Retrieved July 7, 2012 from <http://www.merriam-webster.com/medical/introversion>.
- Judging. (2012). In Merriam-Webster online Dictionary. Retrieved Jul 7, 2012 from <http://merriam-webster.com/dictionary/judging>.
- Keirse, D. (2012). Overview of the four temperaments. Retrieved on July 8, 2012 from http://www.keirse.com/4temps/overview_temperaments.asp
- Keirse, D., & Bates, M. (1978). *Please Understand Me: Character & Temperament Types*. Del Mar, CA: Prometheus Nemesi Books.
- Luthans, F. (2011). *Organizational Behavior, An Evidence-Based Approach (12th ed.)*. New York: McGraw Hill Irwin.
- MacIntyre, A. C. (1981). *After Virtue: A Study in Moral Theory (1st ed.)*. Notre Dame, IN: University of Notre Dame Press.
- McConnell, S. (1999). *After the Gold Rush: Creating a True Profession of Software Engineering*. Redmond, WA: Microsoft Press.
- Merrill, D. W., & Reid, R. H. (1981). *Personal Styles and Effective Performance*. Radnor, PA: Chilton Book Co.
- Millette, V., & Gagné, M. (2008). Designing volunteers' tasks to maximize motivation, satisfaction and performance: the impact of job characteristics on volunteer engagement. *Motivation & Emotion*, 32(1). 11-22.
- Open Advocacy (2012). Why do people create open source software? *How Open Source Works*. Retrieved June 14, 2012 from <http://www.openadvocacy.org/answers/why-do-people-create-open-source-software/>.
- Open-source software (OSS). (2012, June 8). In *Wikipedia, The Free Encyclopedia*. Retrieved July 2, 2012, from http://en.wikipedia.org/w/index.php?title=Open-source_software&oldid=496613745.
- Rhodes, T. (2008). Why do people make software for free? *The Open Source Advocate*. Retrieved June 14, 2012 from <http://www.useopensource.blogspot.com/2007/08/why-do-people-make-software-for-free.html>.
- Sensing or Intuition. (2012). The Myers & Briggs Foundation. Retrieved July 7, 2012 from <http://www.myersbriggs.org/my-mbti-personality-type/mbti-basics/sensing-or-intuition.asp>
- Thinking. (2012). In Merriam-Webster online Dictionary. Retrieved July 7, 2012 from <http://www.merriam-webster.com/dictionary/thinking>.
- TRACOMM Group. (2012). The four social styles —analytical style. Retrieved July 10, 2012 from <http://www.tracomcorp.com/training-products/model/Analytical-Style.html>
- Von Krogh, G., Haefliger, S., Spaeth, S., & Wallin, M.W. (2012). Carrots and rainbows: motivation and Social development in open source software practices. *MIS Quarterly*, 36(2). 649-676.
- Wheeler, D. A. (2007). Why Open Source Software/Free Software (OSS/FS, FLOSS, FOSS)? Look at the numbers! Retrieved July 9, 2012 from http://www.dwheeler.com/oss_fs_why.html#challenges.