

March 2004

Resistance as Motivation for Innovation: Open Source Software

Joseph F. Kavanagh

University of Arkansas, jkavanagh@walton.uark.edu

Follow this and additional works at: <https://aisel.aisnet.org/cais>

Recommended Citation

Kavanagh, Joseph F. (2004) "Resistance as Motivation for Innovation: Open Source Software," *Communications of the Association for Information Systems*: Vol. 13 , Article 36.

DOI: 10.17705/1CAIS.01336

Available at: <https://aisel.aisnet.org/cais/vol13/iss1/36>

This material is brought to you by the AIS Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Communications of the Association for Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.



RESISTANCE AS MOTIVATION FOR INNOVATION: OPEN SOURCE SOFTWARE

Joseph F. Kavanagh
Information Systems Department
University of Arkansas
jkavanagh@walton.uark.edu

ABSTRACT

Resistance is frequently viewed as a negative aspect of human interaction. Although resistance manifests itself in numerous ways, resistance to change is frequent when individuals are introduced to new ideas or innovations. This form of resistance can limit forward progress of either an individual or an organization. However, a few papers investigated possible positive roles of resistance in human life. This paper proposes that resistance can be a positive motivator to achieve change. Open source software (OSS) is a technological innovation that is laden with aspects of resistance. One of the initial motivations for the development of open source software was “psychological reactance” on the part of a few software developers. Reactance is a limited part of the overall construct of resistance; specifically, it refers to resistance is caused by external threats to an individual’s freedom of choice, which generally manifests itself affectively.

This paper looks at the role of resistance as a motivator for technological innovation from the perspective of open source software development. It also presents techniques for overcoming resistance to the adoption of open source software. Specific techniques presented are the Alpha and Omega strategies for overcoming resistance. Alpha strategies work by attempting to increase the approach forces towards some goal. Conversely, Omega strategies attempt to decrease the avoidance forces, thereby removing resistance to change. Both techniques are used in the context of open sourced software development to motivate participants.

Keywords: alpha strategies, free software, open source software, reactance, resistance, technological innovation, omega strategies

I. INTRODUCTION

Resistance is frequently viewed as a negative in the psychological literature, something that should be avoided or minimized. Many attempts to deal with and overcome resistance are described in the literature¹ Beginning around 2000, resistance has been looked at in a more favorable light [Arkowitz, 2002; Beutler, Moleiro, and Telebi, 2002; Piderit, 2000].

¹ The following references contain discussions of overcoming resistance:[Arkowitz 2002; Beutler, Moleiro, and Talebi, 2002; Brehm and Sensenig, 1966; Brehm, et al., 1966; Burger, 1986; Cialdini, et al., 1978; Cialdini, et al., 1975; Dal Cin, Zanna, and Fong, 2004; Davis and Knowles, 1999; Freedman and Fraser, 1966; Haugtvedt, et al., 2004; Knowles and Lin, 2004a, 2004b, and

Often overlooked is the role that sometimes-negative emotions, including resistance, play in motivating human behavior. Knowles and Lin [2004a] characterize psychological resistance broadly:

“Resistance has acquired a dual definition in psychology. On the one hand, it defines an outcome: The outcome of not being moved by pressures to change. On the other hand, it identifies a motivational state: The motivation to oppose and counter pressures to change” [p.5].

The term reactance is related to resistance in that reactance theory preceded the theory of resistance, and the term “resistance” is generally considered to encompass reactance, even though reactance is only a small part of the whole. Reactance represents the affective component of resistance. Unfortunately, the definition of both reactance and resistance are somewhat loose and is often used interchangeably or incorrectly in the literature [Knowles and Lin, 2004a].

An innovation is defined as an idea, practice, or object that is perceived as new by an individual or other unit of adoption and by its nature, creates change [Rogers, 1983]. As such, it has the potential to evoke resistance. Frequently, in an organizational setting resistance stymies positive change. It can motivate creativity or cause an individual to change the nature of their world. This response is hinted at, within the therapeutic context, by Arkowitz [2002] when he states “change implies the possibility of an improved quality of life, but it also implies unpredictability and increased anxiety” (p.222). If the motivation to change the status quo is higher than the negative factors, radical changes are possible.

This paper proposes that instances of resistance spawning technological innovation must occur. That is, in these cases resistance to the status quo is be a powerful motivating factor. The paper uses the instance of a technological innovation, specifically open source software, to look at positive technological change motivated in part by resistance.

The paper is organized as follows: In the next section the background of free and open source software is defined and introduced. In Section III, both the distinction between free and open source software is explained as is their relationship to one another. Other instances of open source software are introduced in Section IV. The role of narratives as a tool to overcome resistance is introduced and illustrated in Section V. Section VI describes previous research dealing with resistance and Information Systems (IS) professionals. The Alpha and Omega strategies to overcome and use resistance to motivate are introduced, defined, and discussed in Section VII. The discussion section which follows (Section VIII) covers the implications for researchers and practitioners and the limitations of the paper. The conclusion section points to other opportunities in both IS and open source software research.

II. FREE/OPEN SOURCE SOFTWARE AS A PRODUCT OF RESISTANCE

Free/Open source software is an enigma to some. The enigmatic nature of the software comes from the novel approach used to disseminate it. To qualify as open source, the source code must be freely and easily available, in addition to the executable program. In addition, , the user or developer must be able to alter and freely redistribute the altered product (with appropriate citation), if desired. This process is, in many ways, analogous to the peer review process in academia, i.e., knowledge is shared, tailored to the need, and disseminated. Open source is more of a way of sharing knowledge than just a way of releasing software. A feature of open source is that the vast majority of developers are not directly compensated for developing open source software. This lack of payment is one of the strangest of open source software development for some to understand. Open source is about the free exchange of ideas. It draws

2004c; McGuire and Papageorgis, 1962; Messer, 2002; Moyers and Rollnick, 2002; Quinn and Wood, 2004; Sagarin and Cialdini, 2004; Sherman, Crawford, and McConnell, 2004; Wegener, et al., 2004; Wicklund and Brehm, 1968; Zuwerink-Jacks and O'Brien, 2004].

heavily on the philosophy of the free software definition. The specific relationship between free and open source software is covered in more depth in Section III. The term free is used in the following context (from Stallman [2004])

“Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

- *The freedom to run the program, for any purpose (freedom 0).*
- *The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.*
- *The freedom to redistribute copies so you can help your neighbor (freedom 2).*

The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this. “

The emphasis on freedom and choice may be the very reason that open source software development, when viewed as a paradigm, may be fulfilling a deep psychological need for the individuals who create the software. It is a basic tenet of psychology that, when confronted with opposition, a human being often seeks to minimize or counteract that opposition. According to the theory of psychological reactance,

“It is possible for an individual to adopt and hold almost any attitudinal position he wants. ... nevertheless, it is possible to threaten the individual's attitudinal freedom by pressuring him to adopt a particular position. What effect, if any, would such a threat have on the individual's attitude? According to a theory of psychological reactance (Brehm, 1966), the effect could be a boomerang attitude change” [Wicklund and Brehm, 1968, p.64].

Knowles and Lin [2004] further identify three components to resistance as affective, cognitive, and behavioral. “I don't like it!, I don't believe it!, and I won't do it! are the affective, cognitive, and behavioral components of resistance, respectively” [p.5].

STALLMAN'S ROLE

What was to become the Open source software movement was officially born in 1984 when Richard M. Stallman began the GNU Project. GNU is a recursive acronym meaning “GNU's Not Unix”. He also founded the Free Software Foundation (FSF), which latter splintered into the open source movement. The following narrative by Stallman is important to understand the role of resistance in the free software and open source communities:

“I had already experienced being on the receiving end of a nondisclosure agreement, when someone refused to give me and the MIT AI lab the source code for the control program for our printer. (The lack of certain features in this program made use of the printer extremely frustrating.) So I could not tell myself that nondisclosure agreements were innocent. I was very angry when he refused to share with us; I could not turn around and do the same thing to everyone else. Another choice, straightforward but unpleasant, was to leave the computer field. That way my skills would not be misused, but they would still be wasted. I would not be culpable for dividing and restricting computer users, but it would happen nonetheless. So I looked for a way that a programmer could do something for the good. I asked myself, was there a program or programs that I could write, so as to make a community possible once again?” [Stallman, 2003].

This experience became the motivation for publishing his GNU Manifesto [Stallman, 2003], which discusses a vision and plan for “free” or open source software. This document ultimately influenced the creation of a community of developers and users who pride themselves on innovation, as foretold by Stallman when he asked himself what could he do to “make a community possible once again”. This experience and the resulting manifesto was also the impetus for the GNU General Public License [Stallman, 1991], a software license that guarantees the user of the software certain rights and freedoms. All subsequent activities of both the free software foundation and the open source software movements build upon the product of this resistance, as evidenced by the use of the GNU General Public License (GPL) or one of its approved derivative licenses² by nearly all free and open source software products. The preamble of the GPL license states:

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too. When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things [Stallman, 1991].

As a direct result of Stallman's resistance to closed proprietary software, the Free Software Foundation created a new type of software paradigm, complete with a previously unseen type of license, the GPL. This license guarantees that all users obtain certain rights and responsibilities. Without this document, there would be no large scale sharing of source code, nor would the user of the software obtain the right to make changes to, or learn from, the source code of others.

The GPL was also an opportunity to garner support from others who held similar ideas about software freedom and sharing. Initially, when commercial computers first were sold, fees for software were the exception rather than the rule. Most computer programming was done in large university or scientific laboratories. Source code was shared by participants, as part of their scientific and research training [von Hippel and von Krogh, 2003]. For some, the advent of closed proprietary software was the anathema to years of openness. Some of these engineers, scientists, and computer scientists were highly resistant to the change. However, prior to Stallman's creation of the Free Software Foundation in 1984, there was no outlet to air their concerns and no way of returning to the collaborative early days of research. This desire to resist closed software spawned a new generation of software products, which includes the entire realm of open source software. The relationship between free and open source software is discussed in the next section.

III. FREE SOFTWARE GETS A NAME CHANGE

In 1991, Linus Torvalds developed a UNIX compatible kernel, which subsequently became the Linux operating system, based largely on the work of the free software community and relying on the rights that are accorded by the GPL. The Linux operating system itself is also licensed under the GPL, which guarantees that the source code will remain open. Clearly, others were also resistant to the changes going on in the computer community. As a result, these individuals were

² All of the approved derivative licenses also guarantee the user the rights to view and alter the source code associated with the product.

motivated to make changes. Regarding Linux, Eric S. Raymond, the author of *The Cathedral and the Bazaar*, and *Homesteading the Noosphere*, states:

“Linux is subversive. Who would have thought even five years ago that a world-class operating system could coalesce as if by magic out of part-time hacking by several thousand developers scattered all over the planet, connected only by the tenuous strands of the Internet?” [Raymond, 1999].

Born of resistance, free software begins to create resistance. The subversive nature that Raymond talks about becomes a barrier to adoption of free software. An attempt to overcome that resistance resulted in a name change. A point that is well made by Stallman:

“Free software’ makes some people uneasy. people may reject the idea for that. ... ‘open source’ is offered ... to be ‘more acceptable to business.’ “ [Stallman, 2003]

IV. OTHER FORMS OF OPEN SOURCE SOFTWARE

In other instances, when confronted by the thought of less freedom to choose the type, the availability, or structure of software, some members of the open source software movement repeatedly stepped forward to fill a gap. For these individuals, the very existence of commercially available prepackaged software, which hides its internal workings, was anathema. They felt that they lacked the freedom to tailor software to meet their needs. Packaged software may cause a boomerang attitude change. Many of the developers of open source software are well paid to develop software in their professional lives, yet they choose to develop open source software for limited or no financial incentives.

Another specific example of this phenomenon was the creators of the Apache web server. The Apache developers were confronted with a serious problem. The development on the web server that they were previously using, which was created by the National Center for Supercomputing Applications (NCSA), was abruptly halted. Since the NCSA software was in the public domain and numerous additions had been made to the program by volunteers, a group of core developers was assembled to create a common distribution that was named the Apache server. This instance of software was kept in the realm of free software because, as the Apache Group argue at their website, “those who benefit from it by using it often contribute back to it by providing feature enhancements, bug fixes, and support for others in public newsgroups. ... This kind of community can only happen with freeware – when someone pays for software, they usually aren't willing to fix its bugs.” [Apache HTTP Server Project, 2003].

V. USE OF NARRATIVES TO FOSTER INNOVATION

Narratives are a tool to overcome resistance, i.e., they can help influence people to take actions that they would ordinarily resist [Dal Cin, et al., 2004]. In an ironic fashion, narratives play an important role, within the domain of open source software development, as both causing and overcoming resistance. Narratives are frequently used by various parties to gain support for their ideas. The GNU manifesto, which was described in Section II, was the first occasion of narratives used to gain support for Stallman's premises: that software should be freely available and changeable. This viewpoint creates a double approach:

1. Resistance to the idea of proprietary software was created.
2. Resistance to participate freely and change the status quo had to be overcome.

This duality of creating some resistance while overcoming another form of resistance is important to the future of open source development.

This concept of changing the status quo is further seen in the results of studies of what motivates open source software developers. For example, Ghosh et al. [2002], obtained on-line responses

from 2784 Open Source/Free Software (OS/FS) developers. When asked their responses for joining and staying in OS/FS development, approximately 50% stated that they joined because they wanted to share knowledge and skills. This number increased to 67% when asked their reason for staying. For both joining and staying in the OS/FS community, over 30% of the respondents think that software should not be a proprietary good and over 20% wish to limit the power of large software companies [Ghosh et al., 2002]. These responses are the second, seventh, and tenth highest out of the fourteen obtained in the survey. Without the GPL and the derivatives associated with open source software, OS/FS developers would not be able to share their knowledge and skills. Similarly, the percentages indicate that OS/FS developers developed resistance to closed source software products. Narratives are used as an effective tool in disseminating and encouraging resistance to proprietary software, while simultaneously overcoming the resistance to participate for little or no financial incentive. This finding leads to the conclusion that resistance can be a powerful motivation to participate.

Three major participants (Stallman, Torvalds, and Raymond), to differing degrees, achieved celebrity status inside and outside the open source community. Numerous articles, both text and on the Internet have been written about the accomplishments and thoughts of these three key personalities. The result of these narratives is that open source software gained popularity and supporters, including major corporate sponsorship, such as Netscape, Red Hat Linux, and IBM.

Within the narratives, near super-human status is granted to the key players in the open source movement. Glass [1999] states that when confronted by the stories (narratives) about open source that "I felt I was deep in a wonderland of superlatives and social (rather than software) engineering by the time I had finished reading." The overall effect on the participants, however, is pronounced. The resistance to share information and to give freely of their talents is reduced, whereas, for many their resistance to commercial software is increased.

VI. ARE INFORMATION SYSTEMS PROFESSIONALS LESS RESISTANT?

Developers resist the adoption of specific software development approaches and methodologies [Hardgrave, Davis, and Riemenschneider, 2003; Hardgrave and Johnson, 2003; Riemenschneider, Hardgrave, and Davis, 2002]. In fact, the developers with the most experience have been the most resistant. This is consistent with Brehm [1966] who states that the individuals most likely to resist a change are the individuals most capable of resisting. Since open source software development is a software development approach, it is reasonable to assume that developers would be equally resistant to adopting this new approach. Yet, in the case of open source, some of the individuals adopting the approach, are qualified and experienced programmers. who appear to embrace this change.

This occurrence is not isolated. Thousands of open source projects are underway, and the number of users range from a few to millions depending on the specific application [Von Hippel and Von Krogh, 2003, p.209]. The majority of software development projects are staffed by well-paid and highly qualified individuals [Hars and Ou, 2002, p.31]. Cook [2001] shows that the developers who participate are, ironically, the most talented and experienced programmers. Clearly, the motivation to oppose change is somehow removed, thereby changing the overall motivational state of the developer to participate.

Something is radically different about the way open source software is developed that reduces the resistance to use this specific approach. It is an apparent contradiction to the previous software development methodology and software development approach research that shows that more experienced developers tend to be the most resistant to change. I hypothesize reasons in the next section.

VII. ALPHA VERSUS OMEGA STRATEGIES TO OVERCOMING RESISTANCE

Knowles and Lin [2004] define two strategies for overcoming resistance based upon an approach-avoidance model. They name the strategies Alpha and Omega. and define them as follows:

“Alpha strategies promote change by activating the approach forces thereby increasing the motivation toward the goal. In contrast, Omega strategies promote change by minimizing the avoidance forces, thereby reducing the motivation to move away from the goal” [p.119].

The motivation to participate in open source projects and to overcome resistance to adopting a new software development approach appear to work through various Alpha and Omega strategies, which are introduced below. These strategies, as well as the open source innovations that prompted their use are summarized chronologically in Table 1 contained in Appendix I.

ALPHA STRATEGIES

The pertinent Alpha strategies are:

1. make messages more persuasive;
2. add incentives;
3. increase source credibility;
4. provide consensus information; and
5. engage in a norm of reciprocity.

These strategies all overcome resistance to participation and increase resistance to closed “proprietary” software.

Narratives. The use of narratives was discussed in Section V. Narratives are used to persuade software developers to participate in open source software projects and to convince them that adopting open source software will be beneficial to them, even though it potentially lowers their monetary gains.

Incentives. Incentives to adopt the open source software development approach and to participate are based upon standing within the social group. Statements by open source developers, such as the desire to share knowledge, participate in new forms of cooperation, and gain help for problems, hint at the importance of the social group within the open source software development process [Ghosh et. al. 2002]. Software development is often not an individual process. Social processes account for significant amounts of variance in software quality [Sawyer and Guinan, 1998, p. 562]. Social groups play a significant role in the creation of software both by fostering creativity and by support gained by the programmer in producing the finished product. Even though financial incentives may be partially removed, social incentives are increased. This view is consistent with Cialdini's [2000] findings that friendship or respect can be an incentive to reduce resistance.

Source Credibility. Prominent spokespersons for the open source community increase its visibility and its credibility. For example:

- Eric Raymond is president of the Open Source initiative
- Linus Torvalds created Linux
- Prominent companies such as IBM embraced the movement.

Their message that sharing their time, effort, and source code will be beneficial, reduces the resistance to participate.

Consensus. Available research [Hars and Ou, 2002 and Ghosh et al., 2002] shows that the developers who participate and adopt the open source approach are qualified and talented. These developers help provide consensus and expand the credibility of the open source community.

Reciprocity. Because open source software is freely available and can be tailored by a developer, we hypothesize that:

Developers are less resistant to adopting the approach and to giving away their own software.

By participating, developers can seek and obtain help from other developers when confronted with problems that exceed their abilities.

These two factors may set up a reciprocal relationship (reciprocal altruism) "Reciprocal altruism is simply cooperation between two or more individuals for mutual benefit, and it is variously known in the literature as social exchange, cooperation, or reciprocation" [Barkow et al. 1992, p. 169].

OMEGA STRATEGIES

Omega techniques that are applicable to overcome resistance to participation or adoption of open source software are:

1. sidestep resistance,
2. consuming resistance, and
3. use resistance to promote change.

Sidestepping resistance. An effective technique to sidestep resistance is to redefine the relationship. For adoption of open source software, Red Hat Linux, Inc. attempted to define their relationship as not a seller of software but as a knowledge repository for support, training, and certification of computer professionals. With this approach, they successfully overcame resistance to adopting the software. They also solicit developer's participation to increase and extend the functionality of their software. In addition to their paid developers, they accept submissions from outside the organization. This re-framing of their relationship to both developers and consumers sidesteps resistance.

Consuming resistance. Resistance to participation may be further reduced by consuming resistance. By giving developers another more acceptable outlet (within their social group) for their resistance, that is resistance to "closed" commercial software, resistance to participation and adoption may be reduced. Individuals are known to have a finite amount of resources when it comes to resistance [Knowles and Lin, 2004]. If the resistance is consumed by focusing it instead on commercial software, less is available to counter the new form of innovation i.e., the adoption of the open source approach.

Promoting change. The open source model is a paradigm shift within the business community. Resistance to closed, proprietary software is a driver of the open source software innovation. By creating resistance to commercial "closed" software development, resistance to the open source software development approach and resistance to changing the fundamental nature of software development may be reduced. Within this context, it is apparently possible to use resistance to promote change.

VIII. DISCUSSION

Conventional wisdom is that change, especially innovation, can (and does) create resistance. The potential for resistance to spawn an innovative change is examined infrequently. Changing

behavior from a closed system to one of freely sharing and open communication is one such innovative change. This change creates a new paradigm for creating and distributing computer software. In similar circumstances, it can be argued that human beings would be resistant to such a change. It would radically alter their way of operating and even their very way of viewing the development of software. Therefore, the open source software approach should be expected to create resistance. Specific to the idea of resistance is the fundamental question: Why would an individual give away goods and services that have significant value? Such a suggestion should increase resistance more so than reducing it. This question is even more vexing when many of the beneficiaries are people who are well paid or large companies [Lerner and Tirole, 2002, p.198].

Yet it appears that resistance to participate in open source development and adopt open source software is reduced (an Alpha strategy). Resistance as a whole, however, is not eliminated. Our conclusion is that the open source software community was effective in transferring the target of resistance to sources external to the community (an Omega strategy). Thus, within the open source software development community there appears to be a successful blend of Alpha and Omega strategies motivate participants to change.

The beginnings of free/open source software are based upon “psychological reactance” (Section I). Reactance is primarily the affective component of resistance which subsequently triggers the cognitive and behavioral components. Reactance increases motivation.

“it is held, on the basis of reactance theory, that the individual will be motivationally aroused whenever any of his freedoms to engage in various behaviors is eliminated or threatened with elimination” [Wicklund and Brehm, 1968, p.64].

As seen in the role played by Richard Stallman, reactance as a motivation has played an important role in developing of the concept of free/open source software.

In the case of open source software innovation, if the type of reactance experienced by individuals such as Stallman were the only factor, open source software would most likely have ended once the initial affective reaction extinguished itself. The cognitive and behavioral components prolonged and expanded the movement. As individuals begin to value the idea of change cognitively, the behavior associated with that change is easier to perform. In other words, the individual must be able to perceive the change as worth making (cognitively) before they will expend resources to make the change (behaviorally). This sequence of cognition prior to behavior is also seen in the motivational literature. Deci and Ryan [2000] state clearly that “most contemporary theories of motivation assume that people initiate and persist at behaviors to the extent that they believe the behaviors will lead to desired outcomes or goals” [p.227]. Cognitive perceptions are important antecedents to actual behavior.

LIMITATIONS

A limitation of this paper is its narrow focus on resistance's role in open source software development. Resistance is not the sole motivator of participation in this context; however, it is an important motivator. Many projects were started because the developer needed a software tool that is not available. For some projects, such as the Apache Web Server (Section VII), need can be a much stronger motivator than resistance.

IMPLICATIONS

The necessity for research on both the positive and negative repercussions of resistance cannot be understated. Each round of technological innovation offers an opportunity for individuals to be resistant to adopting new technology. The implications of this paper for practitioners and researchers are the assertions that resistance can be worked with and overcome to aid in the adoption of innovations. The offers specific strategies for working with resistance to motivate individuals to make changes that reach beyond the realm of open source software development.

The findings that software developers are resistant to new methodologies and that users can be resistant to new technology makes the adoption of new techniques to overcome resistance important to both the researcher and practitioner. By adopting some of the Alpha and Omega strategies outlined in Section VII, resistance can be changed from a negative aspect that must be tolerated to a positive motivation to aid in the adoption of new ideas.

ACKNOWLEDGMENTS

The author thanks Professors Fred Davis, Bill Hardgrave, and Eric Knowles for their valuable comments on this manuscript.

Extensive quotations are included from the work of Richard Stallman and the Free Software Foundation (www.gnu.org). As shown on their web page, this material is Copyright © 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004 by the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA. Verbatim copying and distribution are permitted worldwide, without royalty, in any medium, provided this notice is preserved.

Editor's Note: This article was received on February 27, 2004 and was published on June 21, 2004. It was with the author for 2 weeks for 2 revisions.

REFERENCES

Editor's Note: The following reference list contains hyperlinks to World Wide Web pages. Readers who have the ability to access the Web directly from their word processor or are reading the paper on the Web, can gain direct access to these linked references. Readers are warned, however, that

1. These links existed as of the date of publication but are not guaranteed to be working thereafter.
2. The contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. The author(s) of the Web pages, not AIS, is (are) responsible for the accuracy of their content.
4. The author(s) of this article, not AIS, is (are) responsible for the accuracy of the URL and version information.

Apache HTTP Server Project. (2003), "About the Apache HTTP Server Project – The Apache HTTP Server Project", Retrieved September 11, 2003, from http://httpd.apache.org/ABOUT_APACHE.html

Arkowitz, H. (2002) "Toward an Integrative Perspective on Resistance to Change, *JCLP/In Session: Psychotherapy in Practice*" Vol. 58(2), 2002, pp.219-227

Barkow, J., Cosmides, L., and Tooby, J. (1995) *The Adapted Mind: Evolutionary Psychology and the Generation of Culture*, New York, NY, Oxford University Press

Beutler, L., Moleiro, C., and Talebi, H. (2002) "Resistance in Psychotherapy: What Conclusions Are Supported by Research" *JCLP/In Session: Psychotherapy in Practice*, Vol. 58(2), 2002, pp.207-217

- Brehm, J. *A Theory of Psychological Reactance*. (1966) New York, NY, Academic Press.
- Brehm, J. and Sensenig, J. (1966) "Social Influence as a Function of Attempted and Implied Usurpation of Choice" *Journal of Personality and Social Psychology*, Vol.4, No.6, pp.703-707
- Brehm, J., et al. (1966) "The Attractiveness of Eliminated Choice Alternative" *Journal of Experimental Social Psychology*, 2, pp.301-313
- Burger, J., (1986) "Increasing Compliance by Improving the Deal: The That's-Not-All Technique" *Journal of Personality and Social Psychology*, Vol.51, No.2, pp.277-283
- Cialdini, R. B., et al. (1978) "Low-Ball Procedure for Producing Compliance: Commitment then Cost" *Journal of Personality and Social Psychology*, 36(5), 1978, pp. 463-476.
- Cialdini, R. B., et al. (1975) "Reciprocal Concessions Procedure for Inducing Compliance: The Door-in-the-Face Technique" *Journal of Personality and Social Psychology*, 31(2), pp.206-215.
- Cook, J. (2001) Open source Development: An Arthurian Legend, in Feller, J., Fitzgerald, B. and van der Hoek, A. (Eds.), "Making Sense of the Bazaar: Proceedings of the 1st Workshop on Open Source Software Engineering", available at <http://opensource.ucc.ie/icse2001/papers.htm>
- Dal Cin, S., Zanna, M. P., & Fong, G. T. (2004) "Narrative Persuasion and Overcoming Resistance" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 175-191). Mahwah, NJ, Erlbaum
- Davis, B. P., & Knowles, E. S. (1999) "A Disrupt-Then-Reframe Technique of Social Influence" *Journal of Personality and Social Psychology*, 76(2), 192-199.
- Deci, E. and Ryan, R. (2000) The "What" and "Why" of Goal Pursuits: Human Needs and the Self-Determination of Behavior, *Psychological Inquiry*, Vol.11(4), pp.227-268
- Freedman, J. L., & Fraser, S. C. (1966) "Compliance Without Pressure: The Foot-in the-Door Technique" *Journal of Personality and Social Psychology*, 4(2), 195-202.
- Ghosh, R., et al. (2002) "Free/Libre and Open Source Software: Survey and Study, Final Report, June 2002" <http://www.informatics.nl/FLOSS/report/> Last accessed Nov. 22, 2003.
- Glass, R. L. (1999) "Of Open Source, Linux, ... and Hype" *IEEE Software*, Jan/Feb
- Hardgrave, B., Davis, F., and Riemenschneider, C. (2003), "Investigating Determinants of Software Developer's Intentions to Follow Methodologies" *Journal of Management Information Systems*, Summer 2003, Vol. 20, No.1, pp.123-151
- Hardgrave, B. and Johnson, R. (2003), "Toward an Information Systems Development Acceptance Model: The Case of Object-Oriented Systems Development" *IEEE Transactions on Engineering Management*, Vol.50, No.3, August 2003, pp. 322-336.
- Hars, A. and Ou, S. (2002) "Working for Free? Motivations for Participating in Open-Source Projects" *International Journal of Electronic Commerce*, Spring 2002, Vol.6, No.3, pp.25-39.
- Haugtvedt, C. P., et al. (2004) "Consumer Psychology and Attitude Change" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 283-297). Mahwah, NJ, Erlbaum
- Jacks, J. Z., & O'Brien, M. E. (2004) "Decreasing Resistance by Affirming the Self" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 235-258). Mahwah, NJ, Erlbaum.
- Katz, D. (1960). "The Functional Approach to the Study of Attitudes" *Public Opinion Quarterly*, 24, 163-204.

- Knowles, E. S., & Linn, J. A. (2004a) "The Importance of Resistance to Persuasion" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 1-13). Mahwah, NJ, Erlbaum
- Knowles, E. S., & Linn, J. A. (2004b) "Approach-Avoidance Model of Persuasion: Alpha and Omega Strategies for Change" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 117-148). Mahwah, NJ, Erlbaum
- Knowles, E. S., & Lin, J. A. (2004c) "The Promise and Future of Resistance and Persuasion" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 301-310). Mahwah, NJ, Erlbaum
- Lerner, J. and Tirole, J. (2002) "Some Simple Economics of Open Source" *Journal of Industrial Economics*, Vol. L, June 2002, No. 2, p. 197-222
- McGuire, W. J., & Papageorgis, D. (1962) "Effectiveness of Forewarning in Developing Resistance to Persuasion" *Public Opinion Quarterly*, 26(1), 24-34.
- Messer, S. B. (2002) "A Psychodynamic Perspective on Resistance in Psychotherapy: Vive la Resistance" *Journal of Clinical Psychology*, 58(2), 157-163.
- Moyers, T. B., & Rollnick, S. (2002). "A Motivational Interviewing Perspective on Resistance in Psychotherapy" *Journal of Clinical Psychology*, 58(2), 185-193.
- Piderit, S. K. (2000) "Rethinking Resistance and Recognizing Ambivalence: A Multidimensional View of Attitudes Toward an Organizational Change" *Academy of Management Review*, 2000, Vol.25, No.4, pp.783-794
- Quinn, J. M., & Wood, W. "Forewarnings of Influence Appeals: Inducing Resistance and Acceptance" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 193-213). Mahwah, NJ, Erlbaum
- Raymond, E., *The Cathedral & The Bazaar, Musings on Linux and Open Source by an Accidental Revolutionary*, (1999) New York, NY, O'Reilly Press
- Riemenschneider, C., Hardgrave, B, and Davis, F. (2002) "Explaining Software Developer Acceptance of Methodologies: A comparison of Five Theoretical Models" *IEEE Transactions on Software Engineering*, Vol.28, o.12, December 2002, pp. 1135-1145
- Rogers, E. M. (1983) *Diffusion of Innovations*, 3d. ed., New York, NY, Free Press,
- Sagarin, B. J., & Cialdini, R. B. (2004) "Creating Critical Consumers: Motivating receptive by Teaching Resistance" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 259-282). Mahwah, NJ, Erlbaum
- Sawyer, S. and Guinan, P.J. (1998) "Software development: Processes and Performance" *IBM Systems Journal*, 1998, 37, 4, pp.552-568
- Sherman, S. J., Crawford, M. T., & McConnell, A. R. (2004) "Looking Ahead as a Technique to Reduce Resistance to Persuasive Attempts" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 149-174). Mahwah, NJ, Erlbaum
- Stallman, R. M., (1991) "The GNU General Public License", <http://www.gnu.org/licenses/gpl.txt>. Last accessed May 30, 2004.
- Stallman, R. M., (2003) "Why ``Free Software" is better than ``Open Source"" <http://www.gnu.org/philosophy/free-software-for-freedom.html> Last accessed December 5, 2003
- Stallman, R. M., (2003) "The GNU Project", <http://www.gnu.org/gnu/the-gnu-project.html>. Last accessed December 5, 2003

Stallman, R. M., (2004) "The Free Software Definition", <http://www.gnu.org/philosophy/free-sw.html> Last accessed March 10, 2004

Ven, A. H. V. D., & Poole, M. S. (1995). "Explaining Development and Change in Organizations" *Academy of Management Review*, 20(3), 510-540.

Von Hippel, E. and Von Krogh, G. (2003) "Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science" *Organization Science*, Vol. 14, No. 2, March-April 2003, pp. 209-223

Wegener, D. T., et al. (2004) "Multiple Routes to Resisting Attitude Change" in E. S. Knowles & J. A. Linn (Eds.), *Resistance and Persuasion* (pp. 13-38). Mahwah, NJ, Erlbaum

Wicklund, R. A., & Brehm, J. W. (1968). "Attitude Change as a Function of Felt Competence and Threat to Attitudinal Freedom" *Journal of Experimental Social Psychology*, 4, 64-75.

APPENDIX I. TIMELINE

Table A1. Summary of Resistance in the Context of Free/Open Source Software.

Timeline and Innovation	Strategy used	Explanation
1984; Richard M. Stallman is unable to obtain source code for printer driver.	Alpha strategy	Psychological Reactance – this action pressures Stallman to adopt a closed source model, he reacts by creating the Found's Free Software Foundation (FSF) and publishes GNU Manifesto. Uses Alpha strategies (add incentives and engage a norm of reciprocity) to garner support, i.e., if you wish to use the software created (incentive) by the free software foundation, you must adhere to its tenets and make derived works available (reciprocity).
1989; The GNU General Public License is copyrighted for the first time.	Alpha Strategy	Codifies license that guarantees software user's rights to view and share source code. By formally creating a license two further Alpha strategies (listed below) are used to overcome resistance to participation and increase resistance to closed "proprietary" software. Increase source credibility - by creating a legal document (the license) the message that open source is a viable alternative to proprietary software is more credible. Make messages more persuasive – the language of the license, specifically its preamble persuades the reader that the rights guaranteed by it are in the user's best interest.
1991; Linus Torvalds builds on software created by the (FSF) to create the Linux Operating System.	Alpha and Omega Strategies	By building on the work of the Free Software Foundation he uses others resistance (an Omega strategy, i.e., use resistance to promote change) to closed source software, as well as the Alpha strategy of engaging a norm of reciprocity, to gain contributors. By adhering to the GPL, Linux may be bundled with software that has already been developed by the Free Software Foundation.

Timeline and Innovation	Strategy used	Explanation
1995; Apache Web Server developed.	Alpha Strategy	Kept in the public domain to gain support for its development, again uses Alpha strategies (add incentives and engage a norm of reciprocity) to garner support. Evidenced in the quote “those who benefit from it by using it often contribute back to it by providing feature enhancements, bug fixes, and support for others in public newsgroups. ... This kind of community can only happen with freeware – when someone pays for software, they usually aren't willing to fix its bugs.” [Apache HTTP Server Project, 2003].
1997; The term Open source software is created to overcome resistance to the term “free software”	Omega Strategy	<p>The term “free software”, as advocated by the Free Software Foundation has produced a level of uncertainty and distrust within the business community, the question of using a product that is contrary to proprietary products raises issues regarding product support. This had the result of producing resistance to use the finished software products and gain business support.</p> <p>This is an Omega strategy for bypassing resistance by sidestepping the issue altogether.</p>
1997 – Present; The use of narratives and spokespersons to gain support for OSS products.	Alpha and Omega Strategies	<p>Netscape chooses to release their browser as an open source product, in part due to presentations by Eric S. Raymond, the president of the Open Source Initiative. Linus Torvalds is given stock in Red Hat Inc. when the company goes public via their IPO.</p> <p>Uses the Alpha strategies of increasing source credibility as well as provides consensus information. By having a prominent spokesperson for the Open Source Initiative and the association of Red Hat Inc. with Linus Torvalds, these entities have attempted to reduce resistance to their respective products and trademarks.</p> <p>Red Hat Inc. also uses the Omega strategy of trying to sidestep resistance by redefining the relationship from that of a software vendor to that of a source of technical support.</p>

ABOUT THE AUTHOR

Joseph F. Kavanagh is a PhD student at the Sam M. Walton College of Business at the University of Arkansas. He spent 12 years in and working for the United States military, primarily working in hospital and medical research environments. His assignments included the Armed Forces Institute of Pathology, the Bureau of Naval Personnel, the National Naval Medical Center, and the Uniformed Services University of the Health Sciences. His interest in open source software includes topics such as user acceptance, development methodology, and developer motivation. Other research interests include the use and validation of agent based modeling techniques in information systems research.

Copyright © 2004 by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints or via e-mail from ais@gsu.edu



Communications of the Association for Information Systems

ISSN: 1529-3181

EDITOR-IN-CHIEF

Paul Gray

Claremont Graduate University

AIS SENIOR EDITORIAL BOARD

Detmar Straub Vice President Publications Georgia State University	Paul Gray Editor, CAIS Claremont Graduate University	Sirkka Jarvenpaa Editor, JAIS University of Texas at Austin
Edward A. Stohr Editor-at-Large Stevens Inst. of Technology	Blake Ives Editor, Electronic Publications University of Houston	Reagan Ramsower Editor, ISWorld Net Baylor University

CAIS ADVISORY BOARD

Gordon Davis University of Minnesota	Ken Kraemer Univ. of Calif. at Irvine	M.Lynne Markus Bentley College	Richard Mason Southern Methodist Univ.
Jay Nunamaker University of Arizona	Henk Sol Delft University	Ralph Sprague University of Hawaii	Hugh J. Watson University of Georgia

CAIS SENIOR EDITORS

Steve Alter U. of San Francisco	Chris Holland Manchester Bus. School	Jaak Jurison Fordham University	Jerry Luftman Stevens Inst. of Technology
------------------------------------	---	------------------------------------	--

CAIS EDITORIAL BOARD

Tung Bui University of Hawaii	Fred Davis U. of Arkansas, Fayetteville	Candace Deans University of Richmond	Donna Dufner U. of Nebraska -Omaha
Omar El Sawy Univ. of Southern Calif.	Ali Farhoomand University of Hong Kong	Jane Fedorowicz Bentley College	Brent Gallupe Queens University
Robert L. Glass Computing Trends	Sy Goodman Ga. Inst. of Technology	Joze Gricar University of Maribor	Ake Gronlund University of Umea,
Ruth Guthrie California State Univ.	Alan Hevner Univ. of South Florida	Juhani Iivari Univ. of Oulu	Munir Mandviwalla Temple University
Sal March Vanderbilt University	Don McCubbrey University of Denver	Emmanuel Monod University of Nantes	John Mooney Pepperdine University
Michael Myers University of Auckland	Seev Neumann Tel Aviv University	Dan Power University of No. Iowa	Ram Ramesh SUNY-Buffalo
Maung Sein Agder University College,	Carol Saunders Univ. of Central Florida	Peter Seddon University of Melbourne	Thompson Teo National U. of Singapore
Doug Vogel City Univ. of Hong Kong	Rolf Wigand U. of Arkansas, Little Rock	Upkar Varshney Georgia State Univ.	Vance Wilson U. Wisconsin, Milwaukee
Peter Wolcott Univ. of Nebraska-Omaha			

DEPARTMENTS

Global Diffusion of the Internet. Editors: Peter Wolcott and Sy Goodman	Information Technology and Systems. Editors: Alan Hevner and Sal March
Papers in French Editor: Emmanuel Monod	Information Systems and Healthcare Editor: Vance Wilson

ADMINISTRATIVE PERSONNEL

Eph McLean AIS, Executive Director Georgia State University	Samantha Spears Subscriptions Manager Georgia State University	Reagan Ramsower Publisher, CAIS Baylor University
---	--	---