

Chapter 13 Examining the Success of Computerization Movements in the Ubiquitous Computing Era: Free and Open Source Software Movements

in M. Elliott & K. Kraemer (Eds.), *Computerization Movements and Technology Diffusion: From Mainframes to Ubiquitous Computing* (p. 359-380), Medford, New Jersey: Information Today, Inc.

Margaret S. Elliott
University of California, Irvine
CRITO
401 East Peltason Drive
Suite 3200 Berkeley Place North
Irvine, CA 92697-4650
(949) 824-6387
melliott@ics.uci.edu

Abstract [A]

Computerization movements are a type of social movement whose advocates focus on computer-based systems as instruments to bring about a new social order. The success of social movements has been defined in various ways: social acceptance, gaining new advantages, creation of new social policies, the implementation of new laws, or a shift in public perception. However, there is a dearth of research in how computerization movements achieve success. In this chapter, I examine the success of the free software movement and the open source movement in mobilizing people and organizations to join their efforts to develop and use free software. Factors contributing to success include strong beliefs in free software, shared opposition to organizations opposing free software, the split of the free software movement into two movements – the free software movement and the open source software movement, ubiquitous access to work, and software artifacts, and evangelistic literature.

Keywords [A]

Social movements, computerization movements, free software movement, open source software, open source software movement, Free Software Foundation, Open Source Initiative, technological action frames, ubiquitous computing

Introduction [A]

Computerization movements (CMs) are a special type of social movement whose advocates focus on computer-based systems as instruments to alter the existing social order. Research has shown that CMs communicate key ideological beliefs to mobilize movement participation through public speeches, written works, popular stories, television shows, and magazine articles (Iacono & Kling, 1996, 2001; Kling & Iacono, 1988). However, there is a dearth of research into how CMs develop over time into successful movements. This chapter examines the success of a new genre of CM in which large-scale computerization is assumed to have taken place, and information and communication technologies (ICTs) facilitate the mobilization of membership. Specifically, I examine how the free software movement and the open source software movement have achieved success in mobilizing participants through their respective movement organizations, the Free Software Foundation (FSF) (<http://www.fsf.org>) and the Open Source Initiative (OSI) (<http://www.opensource.org>), to develop, use, and promote “free” software to such an extent that the technology has reached the diffusion stage (Rogers, 2005). In addition, I discuss how technology of the ubiquitous computing era facilitates the success of these movements. Being connected “any time, any place” provides vast opportunities for evangelism of CM ideology and for global access to CM software artifacts.

CMs differ from other types of social movements such as the women’s movement or the equal rights movement in that a CM’s ideology is centered on the acquisition of *technology* that

can alter a social or work world. The success of typical social movements has been defined in various ways: social acceptance, gaining new advantages, creation of new social policies, or the implementation of new laws (Gamson, 1975). Some argue that the most important societal change that can occur is a shift in public perception (Iacono & Kling, 1996). For example, the movement against drinking and driving and affiliated organizations like Mothers Against Drunk Driving (MADD) have changed public perception about drunk drivers. How, then, can the success of specific CMs be characterized? This chapter attempts to answer that question by drawing upon social movement research (Blumer, 1969; Dawson & Gettys, 1935; Zald & Ash, 1969) to show the stages that the free and open source software (F/OSS) movements have gone through in their evolution from inception to continuing success.

Kling and Iacono (1988) distinguished between a general CM which supports overall computerization of societies worldwide and specific CMs which advocate the adoption of specific computer-based systems. In this way Kling and Iacono (1988) drew attention to the similar trends of specific CMs across many organizations or social settings.¹ Specific social movements have well-defined objectives or goals, a movement organization, recognized leadership, and loosely formed alliances. In the early stages, specific social movements have no clear objective and exhibit impulsive behavior (Blumer, 1969). As a social movement develops, this disbursed disorganized behavior becomes organized, solidified, and persistent. One framework offered to characterize this rise from disorganization to organization is the set of four stages (Dawson & Gettys, 1935): social unrest, popular excitement, formalization, and institutionalization.

In this chapter, I use this framework to characterize the success of the F/OSS movements by tracing their development through these four stages resulting in the adoption and diffusion of

F/OSS on a global scale. To accommodate the unique characteristics of CMs, the dimension “technology adoption” has been added to the formalization stage and “technology diffusion” to the institutionalization stage. In order for a CM to become institutionalized, the CM technology must have a critical mass (Markus, 1987) of people and/or businesses which believe in the benefits of the technology and are actively engaged in its use.

The following elements of a CM contribute to the successful diffusion of the particular technology being promoted by the CM (Iacono & Kling, 2001):

- Technological action frames – multi-dimensional composite understandings that serve as a catalyst for people to invest vast amounts of time and money into the acquisition of new technology serving to frame their ideas about how to use the technology.
- Public discourse – written and spoken rhetoric that develops around the technology found in various discourses: government, scientific organizations, academic institutions, public press, and organizational and professional arenas.
- Organizational practices – ways that organizations (virtual and real) inculcate the use of technological action frames and discourse into everyday practices that result in the acquisition of CM technology in micro-level settings.

In this chapter, I show how the technological action frames of the F/OSS movements contribute to their success and how they correlate with the four stages: social unrest, popular excitement, formalization, and institutionalization.

The chapter is organized in the following way. First the background literature review for social movements and CMs is presented, followed by a brief description of ubiquitous computing. Next, the movement organizations, the FSF and the OSI, are described, followed by

a section on the growth and development of the F/OSS movements through four stages. Finally, we complete the chapter with a discussion and conclusion.

Background [A]

Social movements can be seen as collective enterprises to establish a new order of life (Blumer, 1969). They develop from a sense of unrest. People are motivated to join a movement when they become dissatisfied with some aspect of the social order and hope that the movement will result in a new system or scheme of living. In the beginning, a social movement is amorphous and lacking in form. As the movement takes hold with members, it becomes more organized, with a recognized set of social rules and values, often with a movement organization (Zald & Ash, 1969), and leader(s) who promote the movement through written and oral treatise. As a social movement develops, it tends to become organized, solidified, and persistent. The framework used in this chapter consists of four stages: social unrest, popular excitement, formalization, and institutionalization (Dawson & Gettys, 1935).

Kling and Iacono (1988) argue that computerization of many ways of life in the United States has been influenced by a set of loosely linked CMs, headed by mobilizing ideologies offered by CM activists who are not necessarily direct employees in the computer industry. They distinguish between a general computerization movement in society in which activists proclaim “revolutionary” social changes and specific CMs focused on specific technologies. They compare five specific particular CMs by the ideological content focusing on the following computer-based technologies: urban information systems, artificial intelligence, computer-based education, office automation, and personal computing. Their research differs from previous organizational analyses of computerization because they identify how these CMs cut across society as important sources of mobilizing ideologies for computing activists.

In their study of Internetworking, Iacono and Kling (2001) combine the sociological notion of *collective action frames* (Snow & Benford, 2000; Snow, Rochford, Worden, & Benford, 1986) with the concept of *technological frames* (Orlikowski & Gash, 1994) to coin the term *technological action frames*. Collective action frames are the conscious strategic efforts by groups of people to fashion shared understandings of the world to legitimate and mobilize collective action, whereas technological frames are the notions that people collectively develop about the appropriate use of types of technology. Technological action frames are collective understandings that mobilize people to invest time and money into new technology and form core ideas about how the specific technology works and should or could be used. Such frames work at a macro level to influence micro-social uses in specific settings such as organizations or personal use.

Based on the conceptual model of CMs presented in Chapter One in this book (Elliott & Kraemer, Chapter One, this volume), the outcomes of CMs are varied: continuation, merging with other CMs, fading out, or the derivation of a counter-computerization movement (CCM). The technological anti-utopianism writings associated with specific CMs often spawn a CCM related to that CM's specific computer-based hardware or software. The anti-utopian views of the CCM advocates usually arise in response to a threat from the use and development of the specific technology promoted by a CM. For example, Microsoft Corporation's efforts to downplay the advantages of free software with anti-utopian views could be characterized as a CCM. There is no evidence of a general CCM forming in response to the general CM (Iacono & Kling, 1996; Kling & Iacono, 1988).

The eventual diffusion of CM technology follows the course of several stages: adoption, use, and widespread implementation into organizations and society. See Elliott and Kraemer,

Chapter One, this volume for a detailed account of the varied paths a CM might follow. In this chapter, I show the correlation between the stages of the social movement and the technological action frames which help the F/OSS technology to become diffused worldwide.

Ubiquitous Computing [A]

Ubiquitous computing refers to ICTs that enable us to connect “any time, any place” to a social or work world (see Elliott & Kraemer, Chapter Twenty, this volume for a description and analysis of CMs in the ubiquitous computing era). This field of research has also been known as pervasive computing (Lyytinen & Yoo, 2002), invisible computing, and calm technology (Weiser, 1991). Indeed, ubiquitous computing technology surrounds us from that nagging email we need to read or answer (from work or home) to those missed messages on our cell phones. Increasingly people are working out of home offices using high-speed Internet connections for global communication. Futurists (Mattern, 2002) predict that new technologies of ubiquitous computing will link the real world of objects, products, and supply chain manufacturing items with the virtual world of the Internet or e-commerce – “smart” objects with location sensors will be tracked by businesses and friends and family. The utopian vision of the ubiquitous computing CM paints a picture of excitement and convenience as technology is promoted as melding with our social lives such that we are oblivious to the technology. In fact, this vision does not match with reality, as suggested by Sorensen and Gibson (Chapter Seventeen, this volume) in their study of people using cell phones for professional use. The authors concluded that the two main barriers for widespread mobile and ubiquitous support were “bandwidth and battery life time.”

Researchers, vendors, and magazine writers portray utopian visions of the advantages of being connected all the time, including the elimination of desktop PCs. Futurists predict that

people will use “wearable computing,” with devices becoming part of clothing connecting people to each other via wireless technology. In the US and Europe, mobile phones have become increasingly popular, providing access to the Internet, instant messaging, digital cameras, computer games, video clips, and execution of Java programs (Mattern, 2002). Blackberries (www.rim.com) are becoming popular with business executives and university professors, yet instead of the “calmness” envisioned in ubiquitous computing as predicted by Weiser (1991), research shows that people are becoming addicted to their use (Mazmanian, Orlikowski, & Yates, 2005) and email serves as a constant interruption in everyday routines.

Ubiquitous computing has contributed to the success of the F/OSS movements in providing pervasive computing environments such that people can easily develop and use F/OSS “anywhere, any place” while also spreading the word about the advantages of using F/OSS. For example, the GNUe project (Elliott & Scacchi, forthcoming) prefers the use of Internet Relay Chat (IRC) as its primary means of communication and since its contributors are from all over the world in varying time zones, they hold meetings at various times. Because all volunteers have home computers with high-speed connections, they can “meet” in a chat room to resolve technical issues, to socialize, and to catch up on movement gossip.

Another way that ubiquitous computing has facilitated the success of the F/OSS movements is by enabling the storage and access of downloadable software modules, software development and installation guidelines, and archives of group activity such as recordation of the IRC sessions. Around 1985, the Internet emerged with ARPANET, resulting in online communities exchanging messages on Usenet groups. This was helpful in spreading the word about the FSF and its downloadable software. By the late 1980s, email was beginning to appear

in select scientific and government communities, and more and more people were participating in Usenet groups.

In the 1990s, the advent of the Web enabled virtual communities to evolve into social and work worlds. The 2000s brought the use of advanced technologies for “ubiquitous computing” where people can connect to the Web anywhere through personal computers using Web browsers, personal digital assistants (PDAs) (Allen, Chapter Four, this volume), mobile phones (Sorensen & Gibson, Chapter Seventeen, this volume), and Blackberries (for reading email).

Ubiquitous computing development environments enable F/OSS developers to work on their code or design “anywhere, any place.” F/OSS programmers who use the GNU/Linux operating system (OS) can now easily move from one location to the next by using the Knoppix GNU/Linux distribution. Knoppix is a free/open source real-time Linux CD that boots and runs completely from CD. It includes recent Linux software and desktop environments, with several F/OSS programs such as OpenOffice.org and Mozilla as well as hundreds of other free/open source programs. Users of Knoppix can work on software development on any hardware platform that accepts Linux and use the Knoppix CD in a remote location for “ubiquitous” software development (www.knoppix.org).

Free Software Foundation and Open Source Initiative [A]

Free software refers to software that is open to anyone to copy, study, modify, and redistribute (Stallman, 1999). The FSF, founded in 1985 by Richard M. Stallman (known as RMS in the F/OSS communities), advocates the use of its GNU General Public License (GPL) as a copyright license which creates and promotes freedom. A popular term heard in the free software community is “Think free speech, not free beer.” It is used to emphasize the importance

of the defense of freedom, not just the ideal of promoting software that is free of cost. It also serves as a technological action frame which mobilizes people to join the free software movement. The term “open source” software was coined in 1998 when a group of free software movement advocates became concerned that the term “free software” was anathema to businesses.

There was a series of events that led to the formation and success of the OSI. As the free software movement became more visible in the mid-1990s, other literature began appearing on the rationale for developing and using free software. Eric Raymond wrote an essay entitled “The Cathedral and the Bazaar” in 1997 (Raymond, 2001) in which he explicated the reasons why he believed that open source licenses resulted in software that is of higher quality but costs less than typical in-house software development. This essay became very popular in the programming community highlighting the open source community to a wider audience.

Around the same time, Netscape was in a battle with Microsoft over whose browser would dominate the market – Netscape Navigator or Internet Explorer. After reading Raymond’s essay and hearing that Microsoft would release its Explorer as part of Windows, managers at Netscape made the decision to open up Netscape software for Netscape Navigator 5.0 (announced on January 22, 1998). This decision greatly increased the open source community’s reputation to the world and to the US business community. In 1998, Eric Raymond, Bruce Perens, and Tim O’Reilly held a meeting and decided that in order to encourage business investment in open source software, the free software people needed to do better marketing.

This meeting was the genesis for the OSI whose mission is to 1) promote the pragmatic benefits to the business community of open source software, and 2) create a certification process for free and open source licenses that meet the open source definition. The OSI is a non-profit

corporation dedicated to managing and promoting the Open Source Definition for the good of the community. Suddenly, vendors started paying attention and offering support for Linux, including Oracle, IBM, and Corel. Since 1998, other companies such as HP have supported Linux as an alternative to Unix or Windows. In 2000, the Sourceforge Web site (sourceforge.net) was initiated in 2000 by VA Linux (now supported by the Open Source Technology Group, a wholly-owned subsidiary of VA Software Corporation) and now includes a CVS repository, mailing lists, bug tracking, message forums, tasks management software, Web site hosting, permanent file archival, full backups, and total Web-based administration. In February 2000, this site hosted 2370 projects and 15,060 registered users. In 2006, it was estimated that more than 1 million users were registered, with over 100,000 open source projects listed.

The major difference between the software development processes advocated by the FSF and the OSI is in the licensing requirements. The OSI promotes more liberties with open source licensing than does the FSF. For example, the OSI supports licenses that accept combinations of open source software with proprietary software while the FSF promotes strict adherence to the principles listed in the GPL, which requires software to be redistributed as free software exclusively. The free software movement, through its movement organization the FSF, has spawned a number of free software projects in which software developers advocate and follow the principle of creating and using free software exclusively. As the time of writing, 5,150 projects are listed on the FSF Web site (www.fsf.org). Both the FSF and the OSI are non-profit organizations with the purpose of recruiting and educating people to support free and open source software.

Stages of Computerization Movements from Inception to Institutionalization [A]

Over time, social movements go through developmental stages as the social movement organization and its culture is established. One way to examine success of a CM is to trace its growth from inception to its institutionalization when it is embedded in society. For the purposes of this chapter, we have assumed that once a CM has made it to the institutionalization phase, it has reached a level of success that promotes continued membership. Another assumption is that even though a movement is institutionalized, its goals may not have been fully achieved so it is still an active institution. For example, the free software movement has a goal of transforming all source code from closed to free, but since achieving this is highly unlikely, the free software movement will be an institution for quite some time. In this section I analyze the success of the F/OSS movements in terms of four stages of development (Dawson & Gettys, 1935) and discuss five mechanisms (Blumer, 1969) which social movements utilize as they grow through those four stages: 1) agitation, 2) development of *esprit de corps*, 3) development of morale, 4) formation of an ideology, and 5) development of operating tactics. For each stage the corresponding technological action frame that predominated is discussed.

Stage 1 – Social Unrest [B]

In the initial stages of a social movement, there may be factors in society making people feel uneasy and restless. This makes people susceptible to appeals and suggestions that fuel their discontent. The role of agitation in this phase is very important in mobilizing people to join a movement. The purpose of agitation is to attempt to loosen people's old attachments or beliefs and replace them with new ideas and impulses. An agitator can be an excitable, restless, aggressive individual, or a calm, quiet, and dignified person stirring people through action not

words – for example, Martin Luther King, Jr. was an agitator for the Civil Rights Movement in the 1960s.

The founder and agitator for the free software movement was clearly RMS who, with a group of hackers in the 1970s, worked in the Artificial Intelligence laboratory at the Massachusetts Institute of Technology (MIT). The MIT hackers developed their own OS to replace the Digital Equipment Corporation (DEC) OS for its PDP-10 machine (Raymond, 2001; Williams, 2002). Written in the arcane computer language, assembler code, the MIT hackers called it ITS for “Incompatible Time-sharing System.” In the early 1980s DEC disbanded the PDP-10, leading the way for RMS to start the GNU project with his own version of a UNIX OS called “HURD,” resulting in the many Unix-compatible utilities available today including Emacs, a word processor for Unix. This was in development in parallel with the LINUX kernel OS and formed the basis for the GNU project (cf. www.gnu.org). As chronicled in Stallman’s many essays (Stallman, 2002), he was passionate about people’s right to alter any source code, and this led to the formation of the FSF in 1982.

The social unrest stage of a CM is not always easily identifiable as in a typical social movement that might be trying to reverse a social injustice such as ethnic prejudice. The goal of a CM is to mobilize people to acquire computerization and may not be seeded in widespread social unrest. However, when looking at the incubation period of the free software movement, one can see that its agitator, RMS, had a clear motive in promoting freedom. The technological action frame, “All software should be free,” became RMS’s mantra and is evident in the description of the FSF purpose:

“The Free Software Foundation (FSF), founded in 1985, is dedicated to promoting computer users’ right to use, study, copy, modify, and redistribute computer programs. The FSF promotes the development and use of free (as in freedom) software ---- particularly the GNU operating system(used widely today in its GNU/Linux variant)---) – and free (as in freedom) documentation. The FSF also helps to spread awareness of the ethical and political issues of freedom in the use of software.

Many organizations distribute whatever free software happens to be available. In contrast, the Free Software Foundation concentrates on development of new free software--- – and on making that software into a coherent system which can eliminate the need to use proprietary (18k characters) software (www.fsf.org).”

In one of RMS’s articles, “Free Software is a Matter of Freedom?” (www.fsf.org), he states:

“Society also needs freedom. When a program has an owner, the users lose freedom to control part of their own lives. And above all society needs to encourage the spirit of voluntary cooperation in its citizens. When software owners tell us that helping our neighbors in a natural way is “piracy,” they pollute our society’s civic spirit. This is why we say that free software is a matter of freedom, not price.”

In the early stages of the free software movement, the discourse regarding the ideological beliefs of the FSF was centered on white papers by members of the FSF and focused on a small cadre of dedicated hackers. With the advent of the Web in the early 1990s, membership in the FSF increased dramatically along with the advent of more and more “free” software projects listed on the GNU Web site. The Web provided RMS with another avenue of reaching more converts to spread the word about the importance of using free software as opposed to proprietary software.

Many other articles by RMS explaining his fervent beliefs in free software can be found on the FSF Web site (www.fsf.org). RMS is clearly an agitator who has established himself as one of the key leaders of the free software movement. He has earned several awards for his leadership in promoting free software:

- 1990 – Richard Stallman was awarded a \$240,000 fellowship by the John D. and Catherine T. MacArthur Foundation.
- 2001 – Richard Stallman was awarded \$830,000 as a co-winner of the 2001 Takeda Award for Techno-Entrepreneurial Achievement for Social/Economic Well-Being to honor “the origination and the advancement of open development models for system software – open architecture, free software and open source software.

From 1971 to 1984, the free software movement was localized to RMS and the MIT AI lab hackers who believed in the philosophy of sharing source code with other programmers. Eventually, they parted ways, some hired by proprietary companies to develop software. Finally, in January 1984, RMS resigned from MIT so that he could continue to develop free software without the university laying claims to his software. By 1985, he had developed a substantial portion of the HURD and its accompanying utilities which later became part of the Linux/GNU

package. He then formed the FSF, a tax-exempt charity, to support his software development and that of his collaborators. He personally created a C compiler, debugger, a text editor (the well-known Emacs), and other tools. Once the FSF was founded, the free software movement moved into the next stage of development, popular excitement, where the dominant technological action frame came to be known as “free as in free speech, not free beer.”

Stage 2 – Popular Excitement [B]

RMS and his collaborators were instrumental in recruiting others to join the FSF during the late 1980s and early 1990s. The stage of popular excitement in a social movement is characterized by people meeting to examine their shared beliefs. The goals of the movement are not as random and aimless but become more focused. There are more clearly defined goals about how social change can occur. Here the leader is likely to be a prophet (as in religious movements) or a reformer. During this period, RMS began organizing the FSF into a movement organization with definite goals. He created the GPL, also known as “copyleft,” to ensure that his code would always be freely available and modifiable for others. The GPL specifies that users of the source code can view, change, or add to the code, provided that they make their changes available under the same license as the original code. In the GPL, RMS lists the freedoms that any person should have when using software under the GPL:

- Freedom to run the program for any purpose.
- Freedom to study how the program works and adapt it to their needs.
- Freedom to redistribute copies of the software at will.
- Freedom to improve the OSS program and to distribute the altered version.

- Required distribution of the originating license that specifies the freedoms and rights concerning the preceding properties.

By 1990, RMS and his team of programmers had created a substantial portion of UNIX utilities and all they needed for a complete free OS was the Unix-like kernel. They started work on the HURD system, an OS based on the MACH microkernel architecture (first developed at Carnegie Mellon). However, at the same time, Linux Torvalds, a graduate student at the University of Helsinki, wrote a Unix-like kernel based on a small Unix clone. Torvalds named it Linux and submitted it to several mailing lists and newsgroups for review. Eventually, other programmers started using the code and submitting changes to Torvalds. Then the Linux kernel was combined with GNU by others to form the GNU/Linux OS which is used by millions today.

With the advent of GNU/Linux, more people began using and modifying the software. This availability of an *entire* OS to replace Unix created quite a stir in the free software community and inspired people to begin free software projects that could be executed in the free environment of GNU/Linux. In addition, this set the stage for the OSI to promote the open source software movement with its technological action frame centered on encouraging businesses to use GNU/Linux along with their proprietary software. Thus, the F/OSS movements moved into the next stage of movement development, formalization.

Stage 3 – Formalization [B]

During the formalization stage, a movement becomes more clearly organized with rules, policies, tactics, and discipline. One way that CMs differ from other types of social movements is that for their movements to be successful, people need to adopt the technology being advocated by the CM organizations. During this stage, a CM becomes more organized and

technology adoption must happen with enough people for the next stage to occur – institutionalization. Without the volunteer efforts of programmers to complete the code, the F/OSS movements would not have reached the institutionalization stage. Programmers are the *raison d'être* for the establishment of F/OSS as an institution and their willingness to develop and use the software is essential to the success of the F/OSS movements. Three of the five mechanisms by which a social movement coalesces and becomes organized are relevant to the understanding of the formalization stage of a CM: 1) development of *esprit de corps*, 2) development of morale, and 3) formation of an ideology (Blumer, 1969). These three dimensions are discussed in detail below. During the formalization stage of the free software movement, a new dominant technological action frame emerged: “Businesses can benefit from use of free software.” During the period 1995–2000, the free software movement split into two factions with two separate movement organizations: the FSF and the OSI. Whereas the FSF advocates the development and use of free source code without the use of proprietary software, the OSI supports a new technological action frame that allows for the combination of free and proprietary software. This split in ideology resulted in an increase in businesses investing time and money into the ongoing development and diffusion of F/OSS software.

The Development of Esprit de Corps [C]

Once the FSF was established in 1985, it began to grow steadily, with increased participation by advocates in the development and use of free software. The members of the FSF developed a sense of *esprit de corps* and began spreading the word to the rest of the world. *Esprit de corps* can be thought of as the organizing of feelings which were aroused by agitation in the name of the movement. It is important to the success of the movement because it develops feelings of rapport, intimacy, shared experience, and of forming a select group (Blumer, 1969).

Esprit de corps can be considered to be an organization of group feeling and a form of group enthusiasm. A common theme in free software communities is the importance of contributing to the free software cause as a means of giving something back to the community. For example, during an interview with a key developer from the GNUenterprise (GNUe) project, when asked about motivation for all of his volunteerism in programming and managing the GNUe project, RMS replied that individuals' motivations vary. He and a colleague started out with the idea of creating free software and became "hooked" on the movement for ethical reasons.

In the GNUe project, a prime form of communication is the daily IRC where people work as a team to resolve conflicts, create software, and socialize at the same time. They become friends even if some have never met face to face. In addition, when newcomers join the IRC and offer to help work on the project, they are gladly welcomed into the community without need for verification of references or programming reputation (Elliott & Scacchi, forthcoming). Another important aspect of this formalization stage is the development of morale.

The Development of Morale [C]

To succeed, a movement must have a persistent and fixed loyalty. "Morale can be thought of as giving persistency and determination to a movement; its test is whether solidarity can be maintained in the face of adversity. In this sense, morale can be thought of as a group will or an enduring collective purpose" (Blumer, 1969, p. 17). The loyalty of members of the FSF to the creed of the free software movement is evident in the methods used to produce free software. There are three kinds of convictions associated with morale: 1) conviction of the rectitude of the purpose of the movement accompanied by the belief that the goals of the movement will result in a lofty vision of an improved world, 2) faith in the ultimate goal of the movement, and 3) belief

in the movement's sacred mission (Blumer, 1969). Morale can be built up as a religious faith in the movement's purpose as members view their leader with saint-like properties. In the following example, there are signs of the strong beliefs in free software that help build morale in the free software movement.

In the GNUe case study, my colleague Walt Scacchi and I found examples of two situations where fierce loyalty to RMS and his tenets of free software development were evident (Elliott & Scacchi, 2003a, 2003b). In this example, a programmer refused to use a documentation tool that included a non-free component. The following exchange shows the strength of the convictions people have in using and developing free software. During a two-day chat, one person argued about using non-free software temporarily to read some documentation. He became enraged when no one would agree with his strict adherence and made a reference to RMS by sending a message to the mailing list:

I think it is extremely **** that a GNU project would require me to install non-free software in order to read and modify the documentation.

I mean if I cannot make vrms happy on my debian system then what good am I as a Free Software developer? Is docbook really this much of a pain?

I can build html versions of stuff on my box if this is what we have to do. This just irks me beyond anything. I really shouldn't have to be harping on this issue for a GNU project, but some ppl like to take convenience over freedom and this should not be tolerated.

Then the person who developed the documentation responded with a lengthy email stating at the end that his motivation for doing this work is for the “freedom of my son.”

By the way, Daniel, using/writing Free software is NOT about making RMS happy or unhappy. He’s a great guy and all, but not the center of the free universe, nor the motivating factor in many (most?) of our lives. For me, my motivation to be here is a free future for my son.

The Development of Group Ideology [C]

The ideology of a social movement is an essential element in assuring the persistency and development of a movement. It consists of a body of doctrine, beliefs, and myths (Blumer, 1969). The ideology usually consists of two forms: 1) an erudite and scholarly approach, and 2) a popular appeal to the uneducated and the masses (emotional symbols, phrases, etc.). During the formalization stage of the free software movement, the group ideology was strengthened by RMS’s erudite writings and the availability of these articles on the Web site. In addition, during this stage the popular phrase “Think free speech, not free beer” became a symbol of the free movement ideology. The FSF produced pamphlets and members attended Linux conventions to spread their ideology and to promote membership by manning booths.

In 1998, the OSI split from the FSF with the intent to stir interest from businesses for the use and development of free source code. At the same time Web site technology such as Netscape browsers was made available and both the FSF and OSI were able to design Web sites offering FAQs, downloadable software, chat rooms, mailing lists, extensive philosophical literature, and users’ manuals. The Sourceforge Web site was formed, creating a centralized

location where people can recruit developers to work on free/open source projects. As the two segments recruited more members, their respective Web sites began posting opinion papers on how free and open source are different from one another, even though most open source software projects supported by the OSI use the FSF-recommended “copyleft” license, the GPL (see <http://www.fsf.org/> for details).

For example, on the FSF Web site, the following illustrates how FSF members view open source:

Another group has started using the term “open source” to mean something close (but not identical) to “free software.” We prefer the term “free software” because, once you have heard it refers to freedom rather than price, it calls to mind freedom

On the OSI Web site, the following describes its intent:

The Open Source Initiative is a marketing program for free software. It’s a pitch for “free software” on solid pragmatic grounds rather than ideological tub-thumping. The winning substance has not changed, the losing attitude and symbolism have. See the discussion of marketing for hackers for more. So that it is clear what kind of software we are talking about, we publish standards for open-source licenses. We have created a certification mark, “OSI Certified,” to be applied only to software that is distributed under an open-source license that meets criteria set by the Open Source Initiative as representatives of the open

software community. We intend this mark to become a widely recognized and valued symbol, clearly indicating that software does, in fact, have the properties that the community has associated with the descriptive term “open source.”

(<http://www.opensource.net>)

The FSF and OSI have certainly grown over the last ten years and have established themselves with enough users and contributors to be in the institutionalization stage. The group ideology has spread to international communities.

On the Libervis.com Web site (www.libervis.org), a twenty-year-old engineering major from Croatia started a community center for free software. The mission of the site is:

Libervis.com is an ambitious project of building a free software (aka open source) community center (FOSS center) on the Web by providing a Web infrastructure (forums, blogs, wiki, articles) for seamless communication, discussing and learning about any FOSS related area of interest. This community center welcomes all people interested in free software, whether they are supporters of the pragmatical open source ideology or pure free software ideology, but we take stand on the side of free software movement and want to educate everyone that uses free software and enjoys its benefits about the importance of freedom which stands behind these benefits.

The group ideology for free software has spread around the world and is constantly reaching new people. The free software movement and open source software movement have

reached an institutionalization phase where more and more people are aware of the movements through the public discourse readily available on Web sites and circulated at professional conventions.

Stage 4 - Institutionalization [B]

Institutions consist of cognitive, normative, and regulative structures and activities that provide stability and meaning to social behavior (Scott, 2001). When a CM reaches the institutionalization stage, the technology being advocated by the CM has reached a diffusion stage (i.e. organizations and society are actively pursuing the use of F/OSS for personal and business applications) as outlined in the conceptual model for a CM in Chapter One of this volume.

By the time a social movement has reached the institutionalization stage, it has developed into a fixed organization with dedicated personnel and a business structure to carry out the goals of the movement. Leaders take on the role of administrators during this stage. The role of tactics (Blumer, 1969) is important in this stage of development in cementing the established reputation of the movement and in continuing the recruitment of members. Both the FSF and OSI have myriad sets of documentation available on their respective Web sites. Topics covered include movement philosophy, recommendations for volunteerism, links to related sites for downloading free software, detailed documentation on licensing, variations of the GPL, and contact information for movement personnel such as the board of directors, and potential speakers for hire. The role of tactics is important in this stage to maintain the movement membership and retain the current members. Both the FSF and the OSI have developed sophisticated Web sites to

spread their public discourse, drawing in more and more people with links to personal blogs of RMS and wikis among others.

Discussion [A]

This chapter traces the historical development of the free and open source software movements, using a framework with four stages of development for social movements (Dawson & Gettys, 1935): social unrest, popular excitement, formalization, and institutionalization. Both the free software movement and the open source software movement have mobilized people to join their efforts through various forms of public discourse proclaiming their technological action frames. CMs do not develop in the same fashion as other types of social movements due to the need for technology adoption to occur at sufficient levels in order for the movement to sustain itself and attain institutionalization where the technology has reached diffusion in society and industry. Thus, in this analysis, technology adoption was added to the formalization stage and technology diffusion was added to the institutionalization stage. As part of that technology adoption/diffusion process, the belief in free software often cements a programmer's dedication to the proliferation of F/OSS projects through volunteer work. These beliefs form a shared ideology between the free software movement and the open source software movement. In addition to their basic shared history, the F/OSS movements are pulled together by integrating factors that constitute their culture (Gerlach, 2001): shared ideology and shared opposition.

Shared Ideology [B]

Movement ideology serves participants at two levels – 1) a sharing of basic beliefs or core themes, sometimes articulated as slogans (e.g. “Think free speech, not free beer”), and 2) differing interpretations of and emphases on core themes (Gerlach, 2001). Core beliefs can be

shared because they are ambiguous and flexible, capable of being changed to fit a particular group's situation.

Beliefs form the core of ideologies, and as such are an important motivation for individuals to work on free software projects. Recent studies of individuals' motivations for contributing to F/OSS development show ideological motivation. Intrinsic motivations such as pleasure in programming and identity with the open source community were found to be primary reasons for people to participate in free/open source programming (Hars & Ou, 2002; Lakhani & Wolf, 2005). In the survey of seventy-nine free/open source programmers (Hars & Ou, 2002), 16.5% of respondents rated high on altruism and 30% identified strongly with the open source community or had a kin-like relationship with other open source programmers. In a larger survey of 684 free/open source contributors from 287 projects (Lakhani & Wolf, 2005), 42% strongly agreed and 41% somewhat agreed that the "hacker"² community was their primary source of identity, while 30% believed that all source code should be open. A group of altruistic-type programmers (28.6%) surveyed were motivated by giving code back to the free and open source community.

Shared Opposition [B]

The recognition or perception that there is external opposition to their cause helps to draw diverse movement groups together to expand. The split of the FSF and OSI has resulted in participants with a spectrum of beliefs about how to proceed with sharing software – at the one extreme are those who believe all software should be free (i.e. never use non-free) and at the other are those who believe any combination of free and non-free is acceptable (Elliott & Scacchi, 2003a, 2003b, 2004, forthcoming). However, most participants feel that big businesses like Microsoft should be sharing their source code to large-scale programs such as Windows.

This shared opposition enables the two groups to work together and serves to reinforce their ideology.

Although the two movements split over ideological differences, they are united in their position that big businesses like Microsoft should share their software. Eric Raymond (2001) posted a series of memos written in October 2001 by a Microsoft executive (www.opensource.org) and given to him by a company insider:

Open Source Software (OSS) is a development process which promotes rapid creation and deployment of incremental features and bug fixes in an existing code/knowledge base. In recent years, corresponding to the growth of Internet, OSS projects have acquired the depth & complexity traditionally associated with commercial projects such as OSs and mission critical servers. Consequently, OSS poses a direct, short-term revenue and platform threat to – particularly in server space. Additionally, the intrinsic parallelism and free idea exchange in OSS has benefits that are not replicable with our current licensing model and therefore present a long term developer mindshare threat.

The “Halloween Documents” (named by the free software community because they were written at the end of October) are periodically updated by Raymond whenever he receives more information about Microsoft’s opposition to open source.

Conclusions [A]

The ultimate utopian vision of the free software movement is for all source code to be readily available to all individuals, much like books are available in a public library (i.e.

companies like Microsoft give away Windows source code so that consumers can fix bugs and create new versions themselves, etc.). That goal may never be achieved because both the United States and other developed countries sell software for a profit, and giving away millions of lines of code for “free” is anathema to the typical business model. However, these CMs have made inroads into how software companies do business. Software and hardware vendors such as HP and Sun are beginning to sell hardware with the open source GNU/Linux OS included. In addition, an occupational community (Van Maanen & Barley, 1984) has emerged from these movements (Elliott & Scacchi, forthcoming) where F/OSS developers are hired as paid consultants to implement and maintain F/OSS applications in businesses. Table 13.1 shows a summary of the factors related to the success of the free software movement and OSI.

[Insert TABLE 13.1 Here]

The F/OSS movements are a new genre of CM which assumes mass computerization is available. Through centralized movement organizations located on Web sites, these movements have been successful in recruiting individuals either as users or contributors (e.g. programmers, technical writers, or software designers). Their integrated ideology (belief in free software) and integrated opposition (big business trying to block free software) have fueled both of these movements to proliferate. Other factors leading to their success and growth are recognized and respected leaders of the movements (i.e. agitators), development of a strong morale and *esprit de corps*, and technological advancements of ubiquitous computing environments (Sorensen & Yoo, 2005; Sorensen, Yoo, Lyytinen, & DeGross, 2005) that enable connections to work and home any time, anywhere for global access to a work product (i.e. free or open source project). In order

to generalize these findings to other CMs of the ubiquitous computing era, future research could include the application of this framework to analyze the success or failure of other CM technology.

References [A]

Blumer, H. (1969). Social movements. In B. McLaughlin (Ed.), *Studies in Social Movements: A Social Psychological Perspective*. New York, NY: Free Press.

Dawson, C. A., & Gettys, W. E. (1935). *Introduction to Sociology*. New York, NY: Ronald Press Co.

Elliott, M., & Scacchi, W. (2003a, November 9-12, 2003). *Free software developers as an occupational community: Resolving conflicts and fostering collaboration*. Paper presented at the GROUP '03, Sanibel Island, Florida.

Elliott, M., & Scacchi, W. (2003b). *Free Software: A Case Study of Software Development in A Virtual Organizational Culture* (Technical Report No. UCI-ISR-03-6). Irvine, CA: Institute for Software Research, University of California, Irvine.

Elliott, M., & Scacchi, W. (2004). Free software development: cooperation and conflict in a virtual organizational culture. In S. Koch (Ed.), *Free/Open Source Software Development*: Idea Press.

Elliott, M., & Scacchi, W. (forthcoming). Mobilization of software developers: the free software movement. *Information Technology and People*.

Gamson, W. A. (1975). *The Strategy of Social Protest, 2nd Ed.* Belmont, CA: Wadsworth Publishing.

- Gerlach, L. P. (2001). Structures of social movements: environmental activism and its opponents. In *Networks and Netwars: The Future of Terror, Crime, and Militancy*. Los Angeles, CA: Rand Corporation.
- Hars, A., & Ou, S. (2002). Working for free? Motivations for participating in open-source projects. *International Journal of Electronic Commerce*, 6(3):25-39.
- Iacono, S., & Kling, R. (1996). Computerization movements and tales of technological utopianism. In R. Kling (Ed.), *Computerization and Controversy: Value Conflicts and Social Change, 2nd Ed.* (pp. 85-105). San Diego, CA: Academic Press.
- Iacono, S., & Kling, R. (2001). Computerization movements: the rise of the internet and distant forms of work. In J. A. Yates & J. V. Maanen (Eds.), *Information Technology and Organizational Transformation: History, Rhetoric and Practice* (pp. 93-136): Sage Publications.
- Kling, R., & Iacono, S. (1988). The mobilization of support for computerization: the role of computerization movements. *Social Problems*, 35(3):226-242.
- Lakhani, K. R., & Wolf, R. G. (2005). Why hackers do what they do: understanding motivation and effort in free/open source software projects. In J. R. Feller, S. Fitzgerald, S. Hissam & R. K. Lakhani (Eds.), *Perspectives on Free and Open Source Software*. The MIT Press.
- Lyytinen, K., & Yoo, Y. (2002). The next wave of nomadic computing: a research agenda for information systems research. *Information Systems Research*, 13(4):377-388.
- Markus, M. L. (1987). Toward a 'Critical Mass' theory of interactive media. *Communication Research*, 14(5):491-511.

Mattern, F. (2002). Ubiquitous computing: scenarios for an informatized world.

<http://www.m-lab.ch/pubs/Scenarios.pdf>

Mazmanian, M. A., Orlikowski, W. J., & Yates, J. (2005). CRACKBERRIES: The social implications of ubiquitous wireless e-mail devices. In C. Sorensen, Y. Yoo, K. Lyytinen & J. I. DeGross (Eds.), *Designing Ubiquitous Information Environments: Socio-Technical Issues and Challenges* (pp. 337-344). Cleveland, Ohio: Springer.

Orlikowski, W., & Gash, D. C. (1994). Technological frames: making sense of information technology in organizations. *ACM Transactions on Information Systems*, 12:174-207.

Raymond, E. S. (2001). *The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly & Associates.

Rogers, E. (2005). *Diffusion of Innovations, 5th Ed.* New York, NY: The Free Press.

Scott, W. (2001). *Institutions and Organizations*. Thousand Oaks, CA: Sage Publications.

Snow, D. A., & Benford, R. O. (2000). Framing processes and social movements: an overview and assessment. *Annual Review Sociology*, 26:611-639.

Snow, D. A., Rochford, E. B., Worden, S. K., & Benford, R. O. (1986). Frame alignment processes, micromobilization, and movement participation. *American Sociological Review*, 51:464-481.

Sorensen, C., & Yoo, Y. (2005). Socio-technical studies of mobility and ubiquity. In C. Sorensen, Y. Yoo, K. Lyytinen & J. I. DeGross (Eds.), *Designing Ubiquitous Information Environments: Socio-Technical Issues and Challenges* (pp. 1-16). New York: Springer.

- Sorensen, C., Yoo, Y., Lyytinen, K., & DeGross, J. I. (2005). *Designing Ubiquitous Information Environments: Socio-technical Issues and Challenges*. Cleveland, Ohio: Springer.
- Stallman, R. M. (1999). The GNU operating system and the free software movement. In C. DiBona, S. Ockman & M. Stone (Eds.), *Open Sources: Voices from the Open Source Revolution* (pp. 53-70). Sebastopol, CA: O'Reilly & Associates, Inc.
- Stallman, R. M. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*: GNU Press.
- Van Maanen, J. V., & Barley, S. R. (1984). Occupational communities: culture and control in organizations. *Research in Organizational Behavior*, 6:287-365.
- Weiser, M. (1991). The computer for the 21st century. *Scientific American*, 94-104. Reprinted in *IEEE Pervasive Computing*, Jan.-Mar. 2002, 19-25.
- Williams, S. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software*. Sebastopol, CA: O'Reilly & Associates.
- Zald, M. N., & Ash, A. (1969). Social movement organizations: growth, decay, and change. In B. McLaughlin (Ed.), *Studies in Social Movements: A Social Psychological Perspective*. New York: Press.

Endnotes [A]

¹ For an in-depth discussion of previous CM research, see Elliott & Kramer, Chapter One in this volume.

² A slang term for a computer enthusiast, i.e., a person who enjoys learning programming languages and computer systems and can often be considered an expert on the subject(s). Among professional programmers, depending on how it is used, the term can be either complimentary or derogatory, although it is developing an increasingly derogatory connotation. The pejorative sense of *hacker* is becoming more prominent largely because the popular press has co-opted the term to refer to individuals who gain unauthorized access to computer systems for the purpose of stealing and corrupting data. Hackers themselves maintain that the proper term for such individuals is cracker.