Open Source, a Phenomenon of Generation Changes in Software Development: The Case of Denshin 8 Go

Hideki FUJITA\textsuperscript{a)} and Fumihiko IKUINE\textsuperscript{b)}

Abstract: Success in software development is not simply “a person” or “a team” succeeding in development, but rather the execution of a smooth “shift to the next-generation” of developers, who continuously upgrade a single software product over a long span of time. In this case, different generations of developers share the source code. This paper analyzes this process using “Denshin 8 go,” Japanese free software as one example. Initially, the original creator, Ishioka developed Denshin 8 go single-handed as closed source software and succeeded in motivating users through frequent upgrades. Several years after the initial release of Denshin 8 go Ishioka lagged behind in much needed upgrades, but one user group continued to use Denshin 8 go and was eager to improve the software. He disclosed the source code to the group, which in turn carried forward the process of development. As this case shows, the viewpoint that alternate generation of developers

\textsuperscript{a)} Faculty of Business Administration, Toyo University, 5-28-20 Hakusan, Bunkyo-ku, Tokyo, Japan, fujita-h@toyo.jp
\textsuperscript{b)} Division of Policy and Planning Sciences, Faculty of Engineering, Information and Systems, University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, Ibaraki, Japan, fikuine@sk.tsukuba.ac.jp

A part of this paper was originally published as Fujita and Ikuine (2005) in Japanese.
outside of a single company is caused by success in software development is known as “open source” software development. However, the reverse is not necessarily true. That is, simply becoming open source does not guarantee successful development.

Keywords: open source, free software, intrinsic motivation, generation changes, Denshin 8 go

Introduction

Success in software development does not simply mean the development of superior software in terms of function and stability by “a person” or “a team.” According to Ikuine and Fujita (2013), who hold that continued development provides the ultimate guarantee for software quality, success is the smooth shift to the next-generation of developers to continuously upgrade a single software product over a long span of time. Superior software is merely the result of continuous upgrades.

However, existing research on Open Source Software (OSS) have misinterpreted the meaning of success. This indicates that focus is solely placed on the finished quality of the software and discussions are limited to whether the software is open source. In addition, they contend that unless the software is open source, it will not attract users with a high development competence, and without shared source code, development communities will not be established. They also assert that open source is a truly effective path to successful development and the establishment of development communities naturally makes the software “future-proofed” (Raymond, 1997).

Therefore, the OSS guidebook (Fogel, 2005; Sandred, 2001) does not cite project survival methods when discussing the methods of
Open source, a phenomenon of generation changes in software development

start-ups and management of projects. Research analyzing OSS success factors (Bonaccorsi & Rossi, 2003; Lee, Kim, & Gupta, 2009; Midha & Palva, 2012) only focuses on developer motivation and the continuous users, paying no attention to project survival.\(^1\) Even longitudinal studies on OSS projects (Sen, Singh, & Borle, 2012; Subramaniam, Sen, & Nelson, 2009) suggest that type of licensing agreement is a success factor.\(^2\)

Although the initial development of Denshin 8 go (pronounced ‘denshin hachi-go’), a leading example of Japanese free software, was completely closed source, a user group was handed over the source code and continued the process of development. Because the original creator, Takamitsu Ishioka, was able to succeed in motivating users to cooperate in the development process through early and frequent upgrades, the users continued to submit requests and reports even when Ishioka lost his zeal and lagged behind in much needed upgrades. Ishioka decided to provide the source code to a user group that had continued to use and display eagerness to improve Denshin 8 go. The source code was shared with the user group in accordance with the change in developer generation, that is, from Ishioka to the user group, and although it was incomplete, to a certain extent Denshin 8 go had become open source.

The case of Denshin 8 go demonstrates the following two points: (1) Software development success means the continuation of development and upgrades, of which superior software is merely a by-product. (2) Open source is a phenomenon observed during the shift to a next-generation of developers, which is a process that accompanies continued development. This paper analyzes the process involving a shift in developer generation by examining a case

\(^1\) While research by Wang (2012) covers OSS project survival, the content is not much different from existing research.
\(^2\) They contend that restrictive licenses have a negative impact on the success of OSS, resulting in the reduction of subscribers and developers.
study based on interview surveys with the original creator of Denshin 8 go, Takamitsu Ishioka, and the main members of the user group that “inherited” the source code from Ishioka. The interview data was appropriately supplemented with data obtained from additional email surveys, the Denshin 8 go official website, and release notes.

Case Description: Development of Denshin 8 Go

Denshin 8 go is an email program for Windows that was developed by Takamitsu Ishioka. The volunteer user groups Den8club (pronounced ‘denpachi club’) and Den8dev (pronounced ‘denpachi dev’) took over the development. The program received high marks for functionality and quality and was the recipient of the Fifth Free Software Prize (FSP96), sponsored by the Internet Association of Japan. Denshin 8 go was initially developed as non-open source free software, but the source code is now made public with certain conditions.

Denshin 8 go has the necessary functions of an email user agent, but the size of the program itself is very small, and characteristically boots up and operates smoothly. One of the reasons for its popularity is the absence of a built-in editor and viewer functions, which allows users to use their own preferred editors and viewers.

The program does not have many lines of code; 50,000 lines when Ishioka first developed the program that increased to 100,000 lines of code in 2003. From 1995 to 1996, Ishioka created DDE\(^3\) and publicized it along with the command option. Thereafter, numerous software programs, especially editors, became compatible with Denshin 8 go\(^4\) (Table 1).

---

\(^3\) Dynamic Data Exchange (DDE) allows communication between multiple application programs within the Windows environment.

\(^4\) Currently, several compatible programs have discontinued support and the number of compatible programs introduced on the official website has
Table 1. Examples of cooperating software for Denshin 8 go

| Use for         | Name                     | Author                  | Items |
|-----------------|--------------------------|-------------------------|-------|
| Editor          | Pen-Neko                 | Kazutaka Yamamoto       | 7     |
| Viewer          | Den-Love                 | Satoshi Yamada          | 6     |
| Customization   | Den8 Extended Menu       | Satoshi Yamada          | 3     |
| Coding          | UUDeview for Windows     | Frank Pilhofer & Michael Newcomb | 4 |
| Communication   | SHOWPC for Windows       | KAWA & [Nakama]         | 4     |
| Templates       | Den8 Template Editor     | Jun-Kun                 | 1     |
| Filtering       | Def                      | Rattatta-Shingen        | 1     |
| Mail Handling   | Henshin 8 go             | Satoshi Yamada          | 9     |
| Others          | DenZWrap                 | Satoshi Ohwada          | 6     |
| **Total**       |                          | 30 authors              | 41    |

*Note: Aggregated on March 1, 2003 by authors.*

*Source: http://www.denshin8.jp/links/

Motivation to develop Denshin 8 go

Ishioka was motivated to develop Denshin 8 go owing to a desire for a “user-friendly email program” and the mild motivation to “make money.” Ishioka began using email in 1994 after joining the internet connection service for individuals provided by Internet Initiative Japan. At the time, he found every email program for Windows extremely inconvenient to use. In the spring of 1995, he began thinking “I am going to create a user-friendly email program for Windows.”

He embarked on the development bearing in mind the “Mail Handler (MH),” a popular email program in the UNIX environment, and by the summer of 1995, his software program was up and running. Ishioka began sharing Denshin 8 go with his co-workers and received positive feedback from them. The software stabilized by decreased to seven.
the fall of 1995, after which he submitted Denshin 8 go to the software bulletin board of the internet service provider Asahi Net.

By 1998, Ishioka provided several upgrades per month. His high level of motivation during this period was attributable to his achievement motivation (Atkinson & Birch, 1978) and ability to realize pride in his accomplishment. Responding to requests and reports sent by users and providing new upgrades was not his only source of satisfaction. He was awarded high evaluations on the distribution sites of software and in magazine articles. The number of downloads of Denshin 8 go increased smoothly. These facts brought him joy and pride and encouraged him to continue providing early and frequent upgrades.

There were no test and debug phases in the upgrading process for Denshin 8 go. Ishioka integrated his own desired features, along with requests and reports from users, into the source code and released it as soon as the code was compiled. The users would send requests for new functions and operational reports to Ishioka while using the latest versions. Several years after the release of Denshin 8 go, Ishioka said that he “just listened candidly to what I was told,” and continued to release upgrades in response to almost all users’ requests. When asked by users “can you do it?” Ishioka felt strongly about wanting to satisfy user expectations and was high spirited, thinking “I can do that.”

Ishioka would not ignore the expectations of users. The users submitted numerous requests and operational reports by directly emailing Ishioka, posting to the user mailing list. Among the users was a person who attached the source code for embeddable APOP and MD5 in an email. High-user motivation, posts of praise on Asahi Net, such as “has a nice feel to it,” and reaching the top of the ranking for most downloaded software by a famous software distribution site

---

5 Sensing problems with the elimination of the test and debug phases, from 1996, Ishioka released the beta versions separately.
in Japan, all served as motivation to continue the process of refining the program.

In 1998, however, the frequency of software upgrades gradually declined. While there was no change in the amount of user requests and reports, or in Ishioka’s desire to respond to them, the decrease was due to Ishioka’s perception that “it was sufficient for my personal use.” Furthermore, subsequent Japanese email programs had increased their share, and it did not seem that he would realize his faint hope, held at initial stages of development, of “making money from Denshin 8 go.” In this manner, Ishioka’s enthusiasm for development dwindled, and he began to release version upgrades at a frequency of about once every two months only for serious bugs or for requests that he could personally relate with.

**Ishioka hands Denshin 8 go over to Den8club**

Den8club started operating its mailing list in November 1995. Ishioka also participated in Den8club. Hence, Ishioka continued to receive a substantial number of requests and reports from users but being unable to respond to the requests, felt remorseful about having fallen into a “source-code hoarding” situation. In July 1999, Ishioka incidentally decided to visit the website of Den8club to confirm the presence of users and mailing list managers, who enthusiastically continuing their activities. Impressed with the enthusiasm of users, Ishioka transferred the rights for source code modification and redistribution to Den8club users, given certain conditions.6

Den8club is a volunteer user organization, which is a mailing list managed by Yutaka Kawase, one of the official builders. While Den8club maintains the rights to modify Denshin 8 go and determine redistribution, Den8dev undertakes most of the actual development

---

6 On July 13, 1999, development was transferred on the conditions that (1) a distribution server was secured, and (2) Denshin 8 go would not be fragmented into multiple versions.
activities. Den8dev performed the distribution of unstable alpha versions and the exchange of technical information related to development. There were four official builders whose role is to manage the versions, preventing the fragmentation of Denshin 8 go into multiple versions, creating Denshin 8 go official versions, and organizing user requests and reports.

Before and after the development transfer, Den8club experienced a certain amount of confusion. Yoshihisa Honda, a member of Den8club, reflects “Den8club, which originally came from ordinary group of users was entrusted with the source code, and it was perceived that there was a little confusion.” Furthermore, friction began to develop between Den8club, which had a track record of activities conducted since 1995, and Den8dev, which was newly formed to handle development. People who were dissatisfied with the disclosure of the source code and its conditions left Den8club.

Development process of Den8club and Den8dev

The composition of the development process executed by Den8club and Den8dev largely consists of the following three phases, which is a cyclical process. (1) Patch creation, (2) Integrating patches and translating them into binary files by official builders, (3) Release and reception of requests and operational reports. Users submitted requests and operational reports to the mailing lists of Den8club and Den8dev, after which Yoshihito Obara, one of the official builders, compiled them into a “WishList.” The official builders and the members of Den8dev created patches as per requests and reports that they agreed upon by using the WishList as a reference. There were cases in which the developers created patches to resolve problems, discovered while using Denshin 8 go as users. In addition, there were cases in which patches were created for RFC compliance.

---

7 Assume it as the main members of Den8club congregating to formulate Den8dev.
Open source, a phenomenon of generation changes in software development

and to countermeasure gaps in security.

Official builders, Kenichirou Nakamura and Takahiro Fukui, integrated the patches by referencing the WishList, converting them into a binary format, and then uploading them to the server. New versions with patch integration that had unstable performance and carried the risk of data loss were released as alpha versions only to Den8dev members. Once the risks were eliminated and the program was deemed safe, it was released as a beta version to Den8club members, which was as good as publicly releasing the program. Den8club and Den8dev members were responsible for running performance tests of the alpha and beta versions, and the submission of bug reports and revised patches. Based on this, the official builders reintegrated the patches and fixed the bugs, after which a final version was released to general users.

Early and frequent upgrades of Denshin 8 go

Ishioka and Den8club were able to achieve early and frequent upgrades for Denshin 8 go. Table 2 compiles the frequency of Denshin 8 go upgrades before and after the transfer of development to Den8club. The mean number of days between upgrades during 1995–1997, which is the period before Ishioka lost his enthusiasm, was 25 days with the number of problems solved approximating seven per upgrade. After the transfer of development to Den8club, the frequency fell to 44 days between upgrades with approximately eight problems solved per upgrade. Users were motivated to submit requests and reports because the developers performed early and frequent upgrades. Users submitted more requests and reports in an attempt to receive feedback regarding new versions of Denshin 8 go from developers. Because developers quickly received requests and reports, they were able to immediately begin the development of the next version. Table 2 not only shows the realization of this favorable cycle but also illustrates that Den8club obtained a motivated user
Table 2. The history of Denshin 8 go versions

| Versions | Days  | Problems | Development days per version | Solved problems per version |
|----------|-------|----------|-----------------------------|----------------------------|
| Ishioka 1995–1997 | 33    | 843      | 247                         | 25.545                     | 7.485                     |
| Ishioka 1998–1999 | 4     | 617      | 23                          | 154.250                    | 5.750                     |
| Den8club 1999–2012 | 106   | 4694     | 843                         | 44.283                     | 7.953                     |
| Total     | 143   | 6154     | 1113                        | 43.035                     | 7.783                     |

Note: Aggregated on May 19, 2012 by authors. “Problems” mean the number of new features, specification changes, and debugs.
Source: [http://denshin8.jp/historics.shtml](http://denshin8.jp/historics.shtml) [Hist16.txt][History.txt] attached to the latest package of Denshin 8 go.

base through Ishioka, which is still keeping users motivated.

**Motivational Inheritance from Generation to Generation**

Because the original creator, Ishioka, had lost his drive, it seemed as if the development of Denshin 8 go would come to a halt. However, the volunteer group for Den8club continued to show their conviction through their persistent use of Denshin 8 go and enthusiasm to make improvements through mailing list activities. This fact was instrumental in Ishioka’s decision to disclose to Den8club the source code for Denshin 8 go, which was originally developed using a closed source software platform. The shift in developers from Ishioka to Den8club meant a change in the generation of developers and both parties inevitably ended up sharing the source code. Den8club is continuing the development of Denshin 8 go to this very day.

Denshin 8 go inevitably became open source with the change to a new generation of developers to facilitate the continuation of development. The case of Denshin 8 go shows that being open source
Open source, a phenomenon of generation changes in software development

is not a necessary condition for success of software development. In the case of “Xara Xtreme,” the source code was disclosed for everything but the core rendering library (CDraw), although development was actually discontinued six months after the source code was disclosed. In the case of the “OpenDarwin” project, that went four years after releasing their source code without achieving its goal, Apple Inc. has always managed the latest versions of source code. These two failures were due to problems other than the open source issue and licensing imperfections. Xara Xtreme was upgraded a mere five times, while the OpenDarwin project was unable to even release its own package. Although early and frequent upgrades are effective for strongly motivating users to cooperate during the development process (Fujita & Ikuine, 2013), both Xara Xtreme and OpenDarwin failed because they were unable to achieve this.

After the first release of Denshin 8 go, Ishioka was able to release 33 upgrades in just over two years (Table 2). The continuation of early and frequent upgrades in the early stages of development served to intrinsically motivate users (Deci, 1975), who continued to submit requests and reports even one year after the number of upgrades had dropped. If upgrades are equivalent to external rewards, a decline in the frequency of upgrades means that user motivation also decreases, and thus, the submission of requests and report would likewise decrease. The failures of Xara Xtreme and OpenDarwin can be explained by this concept of extrinsic motivation. However, such was not the case with Denshin 8 go. Ishioka succeeded in intrinsically motivating users with early and frequent upgrades.

---

8 Xara Xtreme is a vector image editor of England-based Xara Company. Although the source code for Xara Xtreme was released in March 2006, version 0.7, released in August 2006, was the last release. http://archive09.linux.com/feature/119790

9 In April 2002, USA’s Apple Inc. released the source code for the operating system that would become the base for Mac OSX. However, unable to create an independent Darwin OS, activities were abandoned in July 2006.
Since the inception of development, Ishioka left the testing and debugging of Denshin 8 go to the users. Furthermore, he refined Denshin 8 go by incorporating into the upgrades almost all of the requests and reports sent by users. Once users obtained the latest version, they realized that Ishioka completely trusted them, encouraging them to increase their contribution to the refinement of Denshin 8 go. Every time Ishioka released a new version, the users acquired information that served as feedback relating to competence and self-determination, which further intrinsically motivated users. Denshin 8 go is a case that exhibits the informational aspect of rewards (Deci, 1975, Proposition III).

The members of Den8club were able to receive feedback through the release of the source code from Ishioka, which was in response to their continual demonstration of motivation even during the period of stagnated upgrades. This feedback became the motivating force for members of Den8club to act as developers. Similar to when Ishioka was undertaking the development, the developers within Den8club continued to motivate users through early and frequent upgrades. Although the frequency of upgrades had somewhat slowed down, the number of issues resolved per upgrade did not fall, which was a recognized success factor in motivating users.

Den8club adopted Ishioka’s development style and entrusted most of the testing and debugging to the users. Requests and reports from users were noted in the WishList, and corrected bugs were listed and disclosed to all users. In addition, Den8club continually gives users information that serves as feedback on competence and self-determination through providing a list of FAQs along with the opportunity for users to introduce self-created home pages on the official site. Not only was the shift to a new generation of developers executed smoothly, but because the new developers kept up with the

---

10 http://www/denshin8.jp/issuetrack/
positive cycle of feedback and motivation, they could continue releasing upgrades for Denshin 8 go, and as a result, Denshin 8 go continues to be steadily refined to this day.

**Conclusion**

Successful software development is not the sudden creation of superior software from the first release by the original creator and development team. It is the ability of a certain piece of software to continually release upgrades, version-after-version over a long period, owing to an alternate generation of developers to successors and new development teams. Continual upgrades are what steadily increase the quality of the software.

Through the shift to a new generation of developers, different generations of developers share the source code. When the alternate generation of developers does not progress within a company or an organization, the source code can be disclosed to developers from other companies or organizations or to the countless developers on the internet. In this case, sharing the source code is known as “open source.” Open source is a phenomenon that inevitably occurs when there is a shift to a new generation of developers and software development is continued over a long span. Successful development and the creation of superior software does not occur simply because the source code is publicly disclosed. The alternate generation of developers observed in the case of Denshin 8 go exemplifies the above argument.

The success of Denshin 8 go was supported by intrinsically motivated users. Because Ishioka was able to succeed in motivating users through early and frequent upgrades, Den8club members continued to use Denshin 8 go and were eager to improve it, even when the frequency of upgrades stagnated. Hence, Ishioka was able to bequeath the source code and entrust the development to
Den8club. Xara Xtreme, OpenDarwin, and Mozilla were unsuccessful in making the shift to a new generation of developers and failed because they were only concerned about the quality of their software, paying little attention to user motivation.

In fact, Raymond (1997) also points out the importance of a shift to a new generation of developers. It is important that “when you [the original creator] lose interest in a program, your last duty to it is to hand it off to a competent successor” and it is necessary to “establish that it was in a safe pair of hands.” The “proof” described by Raymond, is the conscious volition of continuous usage and enthusiasm to improve the software.

There is a likelihood of criticism that users possessing the will and eagerness do not necessarily have a high level of development competence, required to inherit the source code. However, we may consider that if the developer incorporates consecutive user ideas and steadily refines the software, those who perceptively detect this are the very users that possess a high level of development competence. Actually, when Ishioka was developing Denshin 8 go as closed source software, there were users who sent Ishioka the source code that could be directly incorporated into the software. From this, we learn that simply opting for open source software development does not ensure success.

References

Atkinson, J. W., & Birch, D. (1964/1978). *An introduction to motivation* (2nd ed.). New York, NY: Van Nostrand.

Bonaccorsi, A., & Rossi, C. (2003). Why open source software can succeed. *Research Policy, 32*, 1243–1258.

Deci, E. L. (1975). *Intrinsic motivation*. New York, NY: Plenum Press.

Fogel, K. (2005). *Producing open source software: How to run a successful
free software project. Sebastopol, CA: O'Reilly.

Fujita, H., & Ikuine, F. (2005). Datsupakkejika shita sofutouea no kaihatsu: Sofutouea kaihatsu niokeru yuza no soshikika [The development of software free from package: Organizing users in software development]. Akamon Management Review, 4, 51–70 (in Japanese).

Fujita, H., & Ikuine, F. (2013). Free and open source are not necessary conditions of successful development: The case of “Hidemaru-Mail.” Annals of Business Administrative Science, 12, 151–166. doi: 10.7880/abas.12.151

Ikuine, F., & Fujita, H. (2013). Endless development is the best quality assurance: The case of “Hidemaru-Mail.” Annals of Business Administrative Science, 12, 251–263. doi: 10.7880/abas.12.251

Lee, S. Y. T., Kim, H. W., & Gupta, S. (2009). Measuring open source software success. Omega, 37, 426–438.

Midha, V., & Palvia, P. (2012). Factors affecting the success of Open Source Software. Journal of Systems and Software, 85, 895–905.

Raymond, E. S. (1997). The cathedral and the bazaar. Retrieved from http://www.catb.org/~esr/writings/cathedral-bazaar/

Sandred, J. (2001). Managing open source projects. New York, NY: John Wiley & Sons.

Sen, R., Singh, S. S., & Borle, S. (2012). Open source software success: Measures and analysis. Decision Support Systems, 52, 364–372.

Subramaniam, C., Sen, R., & Nelson, M. L. (2009). Determinants of open source software project success: A longitudinal study. Decision Support Systems, 46, 576–585.

Wang, J. (2012). Survival factors for free open source software projects: A multi-stage perspective. European Management Journal, 30, 352–371.

Received August 18, 2012; accepted October 1, 2012