



Cowboys and the Eternal September: Transfiguration of Hacker Aesthetics

Masayuki HATTA^{a)}

Abstract: The concept of a hacker originated in the 1970s, and began to gradually take shape in the 1980s. It began to be discussed actively in various contexts, particularly in relation with the rise of open-source operating systems such as GNU and Linux in the 1990s until the early 2000s. Subsequently, as the times changed, the qualities that were demanded of reputable programmers changed in a major way. Originally, the programmers' capabilities in terms of writing code were most important, while their sociability was less emphasized. However, as computers became more common among the general public, and their social impact increased, hackers were required to be more socially amenable in various ways. Therefore, a large number of open-source projects introduced codes of conduct.

Keywords: hacker culture, organization culture, open source, project management

^{a)} Surugadai University, 698 Azu, Hanno-shi, Saitama, Japan, 357-8555, hatta.masayuki@surugadai.ac.jp
The authors declare there are no competing interests.

A part of this article was presented at the ABAS Conference 2021 Summer (Hatta, 2021).

© 2021 Masayuki Hatta. This is an Open Access article distributed under the terms of the Creative Commons Attribution License CC BY 4.0 (Attribution 4.0 International) license. The CC BY 4.0 license permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Introduction

Our current information society was built by the activities of computer programmers. The concept of a hacker materialized in the 1970s, and gradually took a clearer shape starting from the 1980s. It was voraciously discussed in various contexts, particularly in relation with the flourishing open-source operating systems such as GNU and Linux in the 1990s until the early 2000s. The image of a hacker was akin to an antisocial antihero, rebelling against regulations and bureaucracy.

As the times changed, the qualities that were demanded of reputable programmers changed in a prominent way. Originally, the programmers' capabilities in terms of writing code were most prioritized, while their lack of aptitude in group activities and lower social skills were not emphasized, resorting to "talking via code." However, as computers became more widespread, it became necessary for programmers to become more socially amenable. Despite this phenomenon, the general image of a hacker has remained the same to this day. This study aims to resolve the discrepancy between the current image of hackers and reality.

Hackers as Cowboys

Since the 1980s, the piece of literature that has most influenced society's general view of hackers was probably Steven Levy's *Hackers*, first published in 1984 (Levy, 2010).¹ Interviewing over 140 people, Levy's portrayal of hacker culture and their image defined how society viewed them for many years afterward.²

¹ A second edition was published in 1994, and a 25-year anniversary edition was published in 2010.

² There was criticism that in Levy's portrayal of hackers, he overly simplified the actual various backgrounds of the hackers (e.g., Rolfe, 2016).

The world of computer programmers, originally developed on the foundation of academia centered on colleges and research institutes, allowed for cultivation of their own sets of values, mentalities, and a sense of “aesthetics.” Notably, Levy (2010, pp. 40–46) asserts that the hacker ethics presented as hacker culture and aesthetics is widely known, which he summarizes as follows:

1. Access to computers—and anything that might teach you something about the way the world works—should be unlimited and total. Always yield to the Hands-on Imperative!
2. All information should be free.
3. Mistrust authority—promote decentralization.
4. Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
5. You can create art and beauty on a computer.
6. Computers can change your life for the better.

In addition to the utopian statement that “computers can change your life for the better,” it is noteworthy that the sentiment of anti-authority and anti-centralization is already apparent at this stage.³ Levy portrayed hackers as similar to Robin Hood, who was both a “thief” and a hero, where it was considered noble to counter the authority of computers in restricting freedom, refusing to depend on existing authorities such as nations or universities, and acting on the basis of their own coding abilities. Additionally, the “phone phreaking,” known as a prime example of early “hacking,” was deemed to be unauthorized usage of the long-distance phone system, and thus illegal in the end.⁴ This image of the hacker was reproduced

³ There is a common thread through the current dark image (Hatta, 2020b) and claims like “the right to tinker” (Hatta, 2020a).

⁴ The development of the blue box device that allowed phreaking, later involved Steve Jobs and Steve Wozniak, the founders of Apple (Lapsley, 2013).

in the fields of fiction and entertainment as well (Chandler, 1996) and was often associated with cyberpunk science fiction culture.

Therefore, it is no wonder that the first person to write literature likening hackers to cowboys was Bruce Sterling, one of the founders of the cyberpunk movement in science fiction. Sterling (1994) stated, “(h)ackers long for recognition as a praiseworthy cultural archetype, the postmodern electronic equivalent of the cowboy and mountain man,” calling cyberspace the “electronic frontier,” equating it with the western frontier of the U.S. in the pioneering days. It is also no accident that the name of an organization that continues to this day to defend the rights of digital society is called Electronic Frontier Foundation.⁵ The West of the past was a lawless zone, where everyone had to self-govern. Similarly, the electronic frontier is a world where what is technologically possible is possible, regardless of legality, and is considered to exist according to different criteria than general society.⁶

An easy-to-understand example when considering a hacker-type communication style is Crocker’s Rules, advocated by Lee Daniel Crocker, who developed Wikipedia’s system, MediaWiki (Crocker, 2001). According to Crocker, people’s messages to others should be optimized in terms of information volume, rather than whether they are pleasant for the other. In this etiquette, if one feels offended by a message, they must accept that as their own responsibility. Anyone can call you an idiot, and you can assert that to be a kind act. As asserted by Crocker, when two people adopt Crocker’s Rules, they do not need to paraphrase or socially format their words, which allows for all of the necessary information to be communicated in the

⁵ According to Levy, hackers who code for the common good are present-day Robin Hoods.

⁶ In the Jargon File, Levy-like hacker ethic is designated as 1, while an aesthetic that does not relinquish these types of illegal acts is designated as 2 (Raymond, 2003).

minimum amount of time, reflecting the hacker spirit that emphasizes rationality over feelings.⁷

Cowboys Are Bad People

The image of the cowboy hacker, evaluated only on their coding ability and does not recognize authority, is the inverse of depicting them as lacking cooperation and social skills. If they can be praised for their abilities, they can be caustically lambasted for their lack of ability.

The technical journalist Paulina Borsook pointed out this dark side of hacker culture early on (Borsook, 2001). As the title *Cyberselfish* indicates, Borsook asserted that the tech community culture was selfish, libertarian, anti-government, and anti-regulation in character (especially on the West coast of the U.S.). She said that, however, this pursuit of freedom also includes “the freedom to be an asshole.” Moreover, an overwhelming majority of the tech community at the time were highly educated white males, who tended to scorn women who were thought to be inferior in technical skills, as also indicated by Borsook. Hackers were originally a very uniform segment made up of white, male engineers graduating from top universities, the so-called “bro culture,” evaluating abilities in absolute terms and encouraging a culture that disrespected women and minorities.

Cowboy Coding

It has gradually come to light that the cowboy mentality introduces both advantages and disadvantages to the efficiency of software

⁷ The Jargon File also states, “Hackers have relatively little ability to identify emotionally with other people.” (Raymond, 2003)

development. While the origin of the term is unclear, there is a software development style called “cowboy coding.” According to Ward’s Wiki, cowboy coding is a software development where programmers have near autonomy. The programmer determines the development schedule, programming language, algorithms, development tools, development framework, and coding style, such that one can state to have no official software development management in this process. As a result, cowboy coding presents the following four issues.

1. Lack of release structure: In corporate product development, the product’s release schedule and delivery are prioritized. However, because the majority of cowboy coding is done as a hobby rather than as professional work, release schedules are either taken lightly or ignored altogether. Moreover, there are many cases where maintenance is needed after product release in a normal product development cycle, but such maintenance is often neglected in cowboy coding. This can be explained by the mentality that making a release plan and maintaining or fixing existing code (even though it is sufficient for one’s purposes in many cases) is not “fun” work.
2. Inexperienced developers: In many cases, cowboy coding consists of hobbyists and students, etc., who are not dedicated programmers working for a company. Rather, such developers often work either by themselves or in small groups. As a result, they do not follow the necessary procedures for developing a large-scale system, such as a clear system build with careful testing and version management. Therefore, there are certain types of Brooks’ legal issues in which many external resources cannot be invited smoothly when learning costs increase and a project grows.

3. Uncertain design requirements: Cowboy coding usually involves development for the sake of fulfilling the individual demand of cowboy developers. As such, developing something that works is prioritized over a careful, premeditated design. Therefore, when administering a larger-scale system, the programmers are plagued with scalability issues and design deficiencies such as inconsistent user interfaces.
4. Incompleteness: Once a cowboy developer's personal needs are satisfied, that project basically becomes unnecessary. In that case, the programmers often lose motivation to continue development.

This kind of cowboy coding was at least compatible with the initial open-source development. For the time being, creating and “scratching the itch” (Raymond, 2001) was a developer's primary motivation, and many projects naturally started on a very small scale. Open-source licensing also freed the developer from maintenance obligations, by enacting a “no guarantee” condition. Even with a Full Scratch program that comes with defects in an initial design, the programming improves in its artfulness through that experience, and is not just a setback. Linux, which experienced an almost Full Scratch at least twice, is a good example. The image of a traditional hacker is very compatible with a cowboy.

Cowboys Meet the Eternal September

One of the first cases that revealed the issues with traditional hacker culture was that which is currently known as the “Eternal September.” “Eternal September” is Internet slang for September 1993, specifically. In 1993, AOL officially entered the internet provider market, causing the number of internet users to

skyrocket. Until then, the month of September in which students start their college terms in the U.S. was the only month that “new users” went on the internet and learned the “netiquette” of internet users largely made up of professionals and college students initially, but that period extended to the rest of the year, thus being dubbed “Eternal September.”

The old-timers generally dismissed and disregarded the new users (newbies) in response to the popularization and commercialization of the internet. This reaction did not necessarily come from elitism, but from a mainstream perception at the time where users were simply anxious that the more users there were, the likelier it would be for Usenet to break (Vincent, 1995). The general users coming from AOL, etc., were different people; thus, the main focus became to accept those users experienced with the original culture, rather than forcing integration of the newbies. In other words, there was rampant tribalism where newbies who joined communities were met with an onslaught of abuse (Smith, 2020).

As tribalism establishes a uniform organizational culture and sense of purpose, it is not always negative; yet, excluding new members is disadvantageous to the growth of the community for open source that features acquisition competition of developers and users. This was known experientially from an early stage, but quantitative research on it has also emerged in recent years. For example, Halfaker, Geiger, Morgan, and Riedl (2013) quantitatively analyzed the growth of the English version of Wikipedia, and discovered that in many online communities, there is a pattern where even if there is an increase in the inflow of people despite there not being much correlation between the number of achievements and number of users and developers initially, development stagnates and sometimes decreases. Moreover, there is preliminary research that shows that there is more productivity in open-source projects with more variety in gender and background (Vasilescu, Posnett, Ray, van

den Brand, Serebrenik, Devanbu, & Filkov, 2015).

Taming Cowboys by Code of Conduct

The qualities of hackers have not necessarily changed that much in the last 30 years, but there have tended to be more disadvantages than advantages in the efficiency of software development when incorporating something considered beautiful in the past, with the changing times. To resolve this issue, there is a movement to introduce Codes of Conduct. Codes of Conduct do not have the legal enforceability (so to speak) of a software license based on a copyright, but can determine the appropriate standards of behavior as a member of the community and authorize the release of a violator from a mailing list, etc., in the worst-case scenario.

Codes of Conduct are not necessarily welcomed with open arms when introduced. Developers accustomed to the old style sometimes saw Codes of Conduct as putting shackles on one's "freedom of speech," thus stagnating development. The Code of Conduct that integrated the above type and the concerns of the old-timer community resulted in the introduction of a Code of Conduct in the Linux kernel system.

A Code of Conduct was introduced in a dev project on the Linux Kernel in 2018. Linux creator Linus Torvalds discontinued Linux kernel development "to apologize for past conduct and improve conduct," only 30 minutes after signing this Code of Conduct (Prakash, 2018). Further, fierce debate sparked because the creator of this Code of Conduct had caused a scandal in the past.

The original intent of introducing a Code of Content to Linux kernel development had been to "make the kernel development community a more accessible environment." This Code of Conduct pledges, "In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our

project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation” (The Kernel Development Community, 2018). Linus Torvalds and Greg Kroah-Hartman (effectively the No. 2 developer of the Linux kernel) signed the Code of Conduct. Major developers like Intel’s Dan Williams and Facebook’s Chris Mason also acted as signatories (Prakash, 2018).

However, the Linux development community up until that point had been operating almost completely counter to this Code of Conduct. There were frequent flame wars, and the conduct of the main developers like Torvalds was extremely aggressive (especially against powerless members). Thirty minutes after signing the Code of Conduct, Torvalds sent an apologetic e-mail about his own behavior to that point (Torvalds, 2018). The fact that he announced a temporary break from development to improve his own behavior, showed that he was toxic to the community up until then.

In the e-mail declaring his temporary retirement, Torvalds wrote the following (Torvalds, 2018).

This week people in our community confronted me about my lifetime of not understanding emotions. My flippant attacks in emails have been both unprofessional and uncalled for. Especially at times when I made it personal. In my quest for a better patch, this made sense to me. I know now this was not OK and I am truly sorry.

What can be gleaned from this message is Torvalds taking preventative measures to avoid being questioned on violating the Code of Conduct regarding previous behavior by apologizing at that moment or being forced to apologize and take a break due to the new Code of Conduct.

Conclusion

Over a quarter century has passed since the word “hacker” has emerged in mainstream media, and the image of the hacker continues to change. Terms like “rock star,” “ninja,” and genius are often used to describe hackers. This originates from those who produce an enormous amount of work themselves, like the “10x programmers,” as there are large disparities in the productivity of programmers (DeMarco & Lister, 2013, Figure 8–2). This is an unchanging characteristic that was already described as of 1975 by Brooks (1995), “programming managers have long recognized wide productivity variations between good programmers and poor ones. But the actual measured magnitudes have astounded all of us” (Brooks, 1995, p. 41). In short, he points to the concept that hiring one excellent programmer is more important and efficient than hiring 100 mediocre programmers. This is the basis for justifying the battle for talent in Silicon Valley that still continues today.

In this study, these kinds of top-notch programmers were called “cowboys.” As summarized in Table 1, the image of an innovative, risk-taking cowboy is a favorable persona when launching a project. As such, the cowboys played a large role in developing information technology. However, the cowboys whose slight social issues were overlooked at the time, as there were no substitutes for their work, have started to be criticized and re-evaluated in recent years. This quality of a hacker has been expressed as “toxic” in recent years (Van Rooij & Fine, 2018). This could be an impediment to growth in full-fledged projects. The aforementioned case of Linus Torvalds, and retirement from the Free Software Foundation, the classic hacker organization of Richard M. Stallman, who was deemed to be a classic hacker by Levy (2010), can be said to be examples of this (Porter, 2019).

Particularly in regard to the first stages of software development,

Table 1. Preferred/Non-preferred Characteristics of Hackers

	Early stage	Matured stage
Cowboys	Innovative	<i>Reckless, toxic</i>
Tamed	<i>Risk-averse</i>	Team players

Note: Italic indicates non-preferred characteristics.

Source: the author.

trial and error and individual strengths are most important, and a cowboy's discretion and the uniform culture worked to their advantage. However, initiation by the old-timers was made more difficult due to the Eternal September. Moreover, as a project matures, the number of people involved increases, such that not only coding but also project management and design become more important. Recently, data has become more important than open source's source code, which already exists in abundance, even meriting the phrase "data is the new oil" (Yonego, 2014). Additionally, there is growing awareness of social issues such as treatment of minorities, especially in the U.S. In these circumstances, cowboys would alienate potential newcomers from projects and impede growth. To combat that, many open-source projects have introduced a Code of Conduct, but opposition remains.

References

- Borsook, P. (2001). *Cyberselfish: A critical romp through the terribly libertarian culture of high tech*. New York, NY: Public Affairs.
- Brooks, F. P. (1995). *The mythical man-month, anniversary edition: Essays on software engineering* (2nd ed.). Boston, MA: Addison-Wesley Professional.
- Chandler, A. (1996). The changing definition and image of hackers in

- popular discourse. *International Journal of the Sociology of Law*, 24(2), 229–251.
- Crocker, L. D. (2001). *Crocker's rules* [Website]. Retrieved from <http://sl4.org/crocker.html>
- DeMarco, T., & Lister, T. (2013). *Peopleware: Productive projects and teams* (3rd ed.). Boston, MA: Addison-Wesley Professional.
- Halfaker, A., Geiger, R. S., Morgan, J. T., & Riedl, J. (2013). The rise and decline of an open collaboration system: How Wikipedia's reaction to popularity is causing its decline. *American Behavioral Scientist*, 57(5), 664–688. doi: 10.1177/0002764212469365
- Hatta, M. (2020a). The Right to Repair, the Right to Tinker, and the Right to Innovate. *Annals of Business Administrative Science*, 19, 143–157. doi: 10.7880/abas.0200604a
- Hatta, M. (2020b). Deep web, dark web, dark net: A taxonomy of “hidden” Internet. *Annals of Business Administrative Science*, 19, 277–292. doi: 10.7880/abas.0200908a
- Hatta, M. (2021, August). *Cowboys and the Eternal September: Transfiguration of hacker aesthetics*. Paper presented at ABAS Conference 2021 Summer, University of Tokyo, Japan.
- Lapsley, P. (2013, February 20). *The definitive story of Steve Wozniak, Steve Jobs, and phone phreaking* [Website]. Retrieved from <https://www.theatlantic.com/technology/archive/2013/02/the-definitive-story-of-steve-wozniak-steve-jobs-and-phone-phreaking/273331/>
- Levy, S. (2010). *Hackers: Heroes of the computer revolution* (25th Anniversary ed.). Sebastopol, CA: O'Reilly Media.
- Porter, J. (2019, September 17). *Richard Stallman resigns from MIT* [Website]. Retrieved from <https://www.theverge.com/2019/9/17/20870050/richard-stallman-resigns-mit-free-software-foundation-epstein>
- Prakash, A. (2018, September 21). *Linux has a code of conduct and not everyone is happy with it* [Website]. Retrieved from <https://itsfoss.com/linux-code-of-conduct/>
- Raymond, E. S. (2001). *The cathedral & the bazaar: Musings on Linux and open source by an accidental revolutionary*. Sebastopol, CA: O'Reilly

Media.

- Raymond, E. S. (2003). *The jargon file, version 4.4.7* [Website]. Retrieved from <http://www.catb.org/jargon/>
- Rolfe, M. (2016). Hacker: Creating the narrative of the digital Robin Hood. In M. Rolfe (Ed.), *The reinvention of populist rhetoric in the digital age: Insiders & outsiders in democratic politics* (pp. 135–164). New York, NY: Springer. doi: 10.1007/978-981-10-2161-9_6
- Smith, E. (2020). *Why the Eternal September mindset needs to end* [Website] Retrieved from <https://tedium.co/2020/10/13/eternal-september-modern-impact/>
- Sterling, B. (1994). *The hacker crackdown: Law and disorder on the electronic frontier*. London, UK: Penguin Books.
- The Kernel Development Community. (2018). *Contributor Covenant Code of Conduct—The Linux Kernel documentation* [Website]. Retrieved from <https://www.kernel.org/doc/html/latest/process/code-of-conduct.html>
- Torvalds, L. (2018, September 16). *LKML: Linus Torvalds: Linux 4.19-rc4 released, an apology, and a maintainership note* [Website]. Retrieved from <https://lkml.org/lkml/2018/9/16/167>
- Van Rooij, B., & Fine, A. (2018). Toxic corporate culture: Assessing organizational processes of deviancy. *Administrative Sciences*, 8(3), 23. doi: 10.3390/admsci8030023
- Vasilescu, B., Posnett, D., Ray, B., van den Brand, M. G. J., Serebrenik, A., Devanbu, P., & Filkov, V. (2015). Gender and tenure diversity in GitHub teams. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, (pp. 3789–3798). doi: 10.1145/2702123.2702549
- Vincent, C. R. (1995, October 25). *A cultural transition: The commercialization of the Internet* [Website]. Retrieved from <https://groups.csail.mit.edu/mac/classes/6.805/student-papers/fall95-papers/vincent-culture.html>
- Yonego, J. T. (2014, July 23). *Data is the new oil of the digital economy* [Website]. Retrieved from <https://www.wired.com/insights/2014/07/data-new-oil-digital-economy/>