SILESIAN UNIVERSITY OF TECHNOLOGY PUBLISHING HOUSE

SCIENTIFIC PAPERS OF SILESIAN UNIVERSITY OF TECHNOLOGY 2024
ORGANIZATION AND MANAGEMENT SERIES NO. 195

STANDARDS IN THE IT INDUSTRY – THE DEVELOPERS' PERSPECTIVE

Mikołaj ROGIŃSKI

SWPS University; mroginski@swps.edu.pl, ORCID: 0000-0003-2854-6809

Purpose: The aim of this article was to present the typology of information technology standards and to explore their importance for programmers.
Design/methodology/approach: The research was exploratory in nature, and based on grounded theory and ethnography. The tool used to collect data were interviews.
Findings: On the basis of the research it is concluded that standards were of utmost importance to the respondents, and were a thing that allowed them to work efficiently.
Research limitations/implications: The conducted research was qualitative and inductive. For this reason, there is limited possibility of making generalizations about the results.
Originality/value: The paper presents important findings that might increase the work efficiency of programmers. Additionally the research was conducted using a relatively uncommon approach in IT and management field (grounded theory, ethnography, qualitative methods, interviews).
Keywords: standards, vendor lock-in, IT, software development.
Category of the paper: Research paper.

1. Introduction

Modern computer science owes its development to a number of elements. The concept of the modern computer was created by Turing and was expressed in a theoretical construct called the Turing machine (van Leeuwen, Wiedermann, 2001), whose task was to process strings of the digits 0 and 1. In turn, the architecture commonly used today dates from the 1940s and is the work of von Neumann (1993), who is known more widely in management as one of the founders of game theory.

In terms of hardware advancement, when viewed from a historical perspective, the significant milestones included the invention of the transistor in 1947, as well as Intel's introduction of the first microprocessor (model 4004) in 1971, which hit the broad market. It is noteworthy that Intel did not perceive the microprocessor as a significant product during its development efforts, and instead focused its resources primarily on the development of the

memory (Aspray, 1997). However, it was thanks to these inventions – in a relatively short period of time – that computers, from calculating machines occupying entire rooms, became devices that could fit on a desk.

In the software dimension, such developments can be cited in abundance. From the creation of the FORTRAN programming language, to the COBOL programming language (which was a language that was less ”mathematized” and more geared toward business applications), to the start of work at Bell Labs on the Unix operating system in 1969. On the other hand, it could also be the release of the VisiCalc product in 1979, which was the first spreadsheet for personal computers (Ceruzzi, 2003).

Many factors have influenced what the broader IT market looks like today – from the number of electronic devices in use, to the range of applications, to the number of users. However, there are two among them that share one common characteristic. These were, namely, the openness: on the one hand, of the architecture of the IBM Personal Computer (PC) (IBM, 2006; Vaughan, 2004), and on the other, of the communications standards on which the Internet operates (Davies, Bressan, 2010; Muller, 2005).

This text presents the subject matter of standardization from two distinct perspectives. The first is an attempt to show how standards were born, how they evolved, and what is the current state of standards in information technology. The second, on the other hand, is based on a self-study in which standards are shown from the perspective of the actors whose work depends very much on them – programmers and information system administrators[1].

2. Theoretical background

Consideration of standards is worth starting by looking at them from a historical perspective. The cradle of standardization was the United States of America, and the origins of the work on industry standards can be traced to the activities of engineers working in industrial enterprises since the second half of the 19th century. At that time they formed self-organizing teams, within which certain requirements for specific components, materials or processes, among others, were established by consensus. Previously, standardization was based not so much on any agreements, but simply on the adoption of a given solution. An example is the Morse alphabet, which – although not officially defined as a standard in any way – functioned as a standard *de facto* in telegraphic communications (Russell, 2014).

An important step in formalizing standardization processes was the establishment of private organizations with this profile. These dealt with, among other things, standards for mining (American Institute of Mining Engineers), or electricity (American Institute of Electrical Engineers). Parallel activities in this area were carried out by the federal government – in the 1830s the Office of Weights and Measures was established, and in 1901 National Bureau

of Standards. These organizations, unfortunately, due to the paucity of funding, were for a long time only a façade (Russell, 2014).

3. Standardization in information technology

Attempts to standardize various areas related to computing have been taking place since the very beginning of its existence in the ”modern” form, i.e. since the 1940s. An example is IBM’s standardization of input/output support in its System/360 series computers, for example. This allowed tape drives or printers to interoperate with multiple models of devices, which greatly simplified the product line and made the offerings more attractive in the eyes of customers (Ceruzzi, 2003).

Research on standards in the IT industry, embedded in the economic sciences, has been carried out since the 1980s, when there was an explosion in their development (David, Greenstein, 1990). In the last 30 years, the approach to standards has been evolving along with the development of this market. Importance is often placed on using – beneficial to end users – open solutions. It is noteworthy that these, among other things, are the foundation of the Internet of Things (Duguid, 2015).

Standards in IT are crucial to the ability to develop any system. The standardization of all their components – hardware as well as software, among others – makes it possible to create universal solutions. This was the case with PCs, which began to dominate the market in the 1980s. The cornerstone here was IBM’s decision to also make the PC’s technical documentation available with the start of PC production in 1981, so that independent suppliers could create hardware extensions for it, as well as write independent software. In addition, it was decided that key components of the PC would be developed by third-party companies – the processor was provided by Intel, and the developer of the operating system was Microsoft with its Microsoft Disk Operating System (MS-DOS). The open architecture meant that – in addition to them – clones of the IBM computer with better performance began to emerge. Competing vendors began to take advantage of the open specification, resulting in a strong feedback loop: widespread PCs – growing software resources – growing user base. In this way, other vendors emphasized that a given computer admittedly was not made by IBM, but you could run (almost) any program written for the IBM-PC on it (Bradley, 2011).

In 1987, a line of computers called Personal System/2 (PS/2) entered the market, featuring many of the solutions commonly copied by others – including a 3.5-inch disk drive and a new type of keyboard and mouse slots. PC-class clones were based on components of various generations of the IBM PC. The PC has become an open standard *de facto* because of its widespread adoption. The phrase ”IBM-PC-compatible” emerged, and the test of whether

a particular PC-clone was IBM PC-compatible was the ability to run the Lotus 1-2-3 spreadsheet and Microsoft Flight Simulator on it (Haigh, 2012).

As noted by Dargan (2005, p. 22-23), it was thanks to such unification, which was put in an institutional framework, that it was possible to develop, among other things, the Linux project (under which the operating system kernel is developed) or the Internet as we know it today.

Linux is an open source software[2] (OSS) development project founded in 1991, which at one point operated in so many variants that its development came into question. It became necessary to establish an organization called Free Standards Group, which developed a common specification referred to as Linux Standard Base (LSB). In time, the LSB was adopted as the ISO 23360 international standard, and the Free Standards Group itself was merged into the Linux Foundation organization (Updegrove, 2007). In the area of computer networks, on the other hand, the role of a standard-setter was taken on by the organization Internet Engineering Task Force (IETF) (Bradner, 2003).

Standardization in the IT industry is a win-win phenomenon for all actors who operate in this market. Hardware suppliers create components that can be combined with components from other manufacturers. Software developers can write a single ”core”, and only customize products for specific hardware architectures, or specific components with which that software is to interact. In general, it can be said that standardization for players in this market reduces barriers to entry. Additionally, as a result of it, the potential customer base grows. End users, in turn, have relatively easy access to new technologies as a result, and potentially their degree of digital exclusion is reduced.

4. Characteristics of standards in information technology

Standards in the IT industry can be divided according to numerous criteria. For example, taking into account the subject matter that is subject to the process, one can distinguish those for hardware, software, data or communication protocols, among others. The following section discusses two classifications. The first distinguishes standards by who their creator is; the second divides them by the license under which they are made available.

5. Official and unofficial standards

According to the first division – which runs along the lines of who is the creator – the following basic types of standards can be distinguished:

1. official – created by standardization organizations,
2. unofficial – created by industry consortia.

Standards belonging to the first group are developed and approved by national and international standardization organizations, which operate at several levels. The lowest are national institutions (e.g., the Polish Committee for Standardization, PKN), the higher are regional ones (e.g., *Comité européen de normalisation*), and at the highest level are international bodies (e.g., International Organization for Standardization, ISO).

Approval processes for individual standards are extremely slow in organizations belonging to this group – usually a process that takes several years. In the highly dynamic IT industry, on the other hand, it is important to act quickly in this area. With this in mind, consortia have been formed that create their own standards – belonging to the second group. They are organizations maintained, most often, by membership fees (members are companies, public institutions, academic institutions, but also individuals) operating in a specific market segment. The standards they develop are usually made available to the public and are not subject to intellectual property protection regulations. A primary example of such a consortium can be the World Wide Web Consortium (W3C), which develops Web standards – including such widely used ones as HTML and CSS (Blind, Gauch, 2008).

Another example of this form of operation is the IETF, which has been operating as an independent organization since 1993 (it was previously supported by the United States Government). It is highly formalized. There is no formal membership or fees associated with it, the work is public and anyone can take part in it (IETF, 2021). However, this does not mean that IT companies do not have a say in the solutions that are worked out. As the data shows, in total, the IETF has produced more than 10,000 documents, in which representatives of more than 2600 companies have participated (Arkko, 2021). However, since everyone contributes as an individual who does not have to give an affiliation, it is likely that this number is underestimated.

Many times these consortia have much more influence over the IT standards used than formal organizations. However, this does not mean that one group of entities operates in isolation from the other. For example, the Institute of Electrical and Electronics Engineers (IEEE) – a consortium dedicated to, among other things, setting IT standards – collaborates with the American National Standards Institute (ANSI) (Blind et al., 2010).

6. Proprietary and open standards

The division described below has become particularly evident since the 1990s. It runs along the type of a license under which the standards are made available. In this case, one can distinguish:

1. proprietary standards,
2. open standards.

The distinction between one and the other is not sharp – there are also intermediate states between which a standard can move. Overall, there are four main classes of standards (Cerri & Fuggetta, 2007):

1. proprietary – protected by patents or copyrights, which might be de facto standards due to their wide adoption,
   a) proprietary undisclosed – the documentation is not made public, and use is possible only under a license granted by the owner,
   b) proprietary disclosed – the documentation is public; use is free or upon payment of a certain fee to the owner,
2. concerted disclosed standards – *de facto* standards whose creators are ”closed” associations of organizations; in formulating them, comments submitted during open consultations were used,
3. open standards (concerted) – created by ”open” groups of organizations, whose members are companies, academic centers, etc.,
4. open standards (*de jure*) – approved by standardization organizations.

In the IT industry, so-called *de facto* standards are relatively common. They are characterized by the fact that they are solutions that are used because of their widespread adoption – regardless of who the developer is. One of the earliest examples in this group was, created in 1969 in Bell’s laboratories, the Unix operating system. Under its control in the 1970s the vast majority of computers in academic and research and development centers operated. Its popularity was due to the fact that it could be run on virtually all hardware configurations in use at the time (West, 2003).

7. Standards in practice – the developers’ perspective

8. Research methodology

The research described was a part of a project aimed to study on the reasons for using open source software, which was analyzed through the lens of the diffusion of innovations theory (Rogers, 1962). The study itself was conducted using qualitative methods. Two complementary approaches were used here: grounded theory (Glaser, Strauss, 2009; Konecki, 2000) and ethnography (Kostera, 1996). Derived from grounded theory, coding allowed structuring the collected data and identifying relevant conceptual categories.

The ethnographic approach, in turn, was used in two dimensions. First, as a ”tool”, thanks to which very rich data on the phenomenon under study was collected. This is because semistructured interviews (so-called anthropological interviews) were used here (Kostera, 2003). On the one hand, this made it possible to collect very rich data. On the other hand, it provided an opportunity to follow new threads that were relevant to the respondents and thus benefit from the so-called serendipity (Konecki, 2004). All but one of the interviews were recorded and transcribed. For the interview that was not recorded, notes were taken. For the purpose of presenting the results of the study, each interviewee was assigned a code, with which the individual quoted statements were marked. These quotes are marked in the text with different formatting.

Secondly, ethnography has been used in an object-oriented sense. The record of the survey is ethnographic, that is, it describes in detail – using what is known as thick description (Geertz, 1973) – the area under study. This description presents the issues under study from the point of view of the subjects themselves.

The study was exploratory in nature. Its purpose was to try to answer the question of what factors influenced the use of free and open source software[3] in IT enterprises. One of the important conceptual categories that emerged during the analysis of the collected data was standards, and specifically open standards.

The research was conducted in small IT companies. This made it relatively easy to reach people at all levels of the organizational hierarchy. A total of 13 interviews were conducted with 12 people. The interviews were conducted, according to the grounded theory methodology, until the thematic categories were saturated – that is, until the value of the subsequent interviews from the point of view of informational value began to diminish.

9. Results and discussion

Standards were one of the main categories raised by respondents during their interviews. As the interviewees claimed, standardization allowed to be independent from the vendor (a phenomenon referred to as vendor lock-in). This was true both for companies, which may have had to incur large costs associated with being ”attached” to specific solutions, but also for users, who may find themselves in an identical situation. With standards comes the security of maintaining business continuity.

The specific topic that interviewees most often referred to was file formats. A format is a standard-compliant way of storing data. Referring to the classifications described in the section 3, it can be described by indicating the type of its creator. It can also be characterized in terms of openness. It is this property that was mainly raised in the interviews.

[Interviewee 2] When a programmer creates code, he does it internally in the company. These are text files simply. The more text files in circulation the better. (…) Because a text file simply contains human-readable content. Well, and possibly formatting. (…) A text file is a text file. Simply put. The ASCII code is inside, publicly known. Alternatively, if you're delivering some files externally somewhere, well, a PDF is a gold standard. That way, some years from now, there will be access to it. Only the more text files the better.

Statements of this type appeared relatively frequently in the course of the survey. The interviewee here referred to several important issues related to standardization and the working environment. IT companies are usually engaged, at least in part of their business, in software development. Even if hardware is produced, the software layer is also usually an important component of it. Software from the programmer's point of view is source code, i.e. instructions written in one of the programming languages that are human-readable[4]. Source code, in turn, is stored in the form of text files, that is, files whose contents are successive lines composed of characters. They are universal due to the fact that they can be read on virtually any hardware. A text file is also considered to be "the lowest common denominator of data storage formats" (Murrell, 2009).

From the programmers' point of view, text files are considered a kind of gold standard. They are seamless to use – they can be opened and edited on virtually any hardware and using a very wide range of editors. In addition, working with them means a relatively high degree of freedom in the choice of hardware and software. Also mentioned in the interview cited above was the so-called ASCII code (American Standard Code for Information Interchange). This is a system for encoding characters in electronic communications, which has been standardized by ISO[5].

The PDF format mentioned in the statement cited above has similar advantages. It has been an open ISO standard since 2008 and, among other things, is a common format for saving text documents (including formatting, but also graphic elements, for example).

In addition to the standards for file formats, and the related comfort of work, interviewees also raised the issue of the tools (software) used. With regard to it, two perspectives were present in the interviews. First, it was important to use software that not only enables specific tasks to be carried out, but that at the same time to be as popular as possible on the market.

[Interviewee 5] Usually, to solve one problem on the Internet, or in the world in general, there are 3, 5, 10 tools. We usually choose one. We try to use solutions like the most standard ones. The most standard, because they are the most tested.

The advantage of this approach was highlighted in the statement quoted above. In the case of closed-source software, testing and quality assurance is largely on the side of the developer of a given solution, and its widespread use is important, though arguably not crucial. In contrast,

the situation is different in the case of open source, community-developed software. Here, widespread use translates into the number of people directly involved in development, which in turn can have a positive impact on quality. This is a reference to Raymond's (1999) formulation of the so-called Linus's Law[6]. It proclaims that *given a large enough beta-tester and co-developer base, virtually every problem will be characterized quickly and the fix obvious to someone*, or in short: *given enough eyeballs, all bugs are shallow*.

Based on the data collected, there are four cases relating to the relationship between software and data storage standards, or file formats. Closed software, i.e., software whose source code is not publicly available, supports closed file formats (case 1), as well as – usually – open ones (case 2). Open source software, on the other hand, usually relies on support for open file formats (case 3). Closed formats in OSS (case 4) are often supported less well. This situation is due to several factors, the most important of which are licensing issues and lack of adequate documentation.

The issue of problems related to the handling of open and closed standards is well summarized in a statement by one interviewee regarding standards in office suite files.

[Interviewee 7] Because you can use closed software and open formats. For example, in Microsoft Office you can save files in ODT format. But it always depends on how (…) the vendor supports this format, that is, whether it saves everything according to the documentation of the format or not. Because I know many programs that, despite the fact that this format is popular, don't save data in compliance with this format.

In fact, there is an internationally defined ISO/IEC 26300 standard, which was established in 2006 (ISO, 2006). It defines OpenDocument, which is an open file standard for office suites that includes text documents, spreadsheets and presentations, among others. Interestingly, a competing ISO/IEC 29500 standard was adopted in 2008 covering exactly the same area (ISO, 2008). The author of the former is the independent consortium Organization for the Advancement of Structured Information Standards (OASIS), while the author of the latter is Microsoft. Thus, there are two standards of the same level – both published by ISO.

From the point of view of the market as such, and the people using them in particular, this is not an ideal situation, although it cannot be said to be unequivocally unfavorable either. The important thing is what the interviewee pointed out in the statement cited above. The problem is not the mere existence of two twin standards. Instead, it is the way they are handled by the software available on the market. This is because they can be implemented in so-called extended versions, so that a file correctly opened and displayed with one tool may not be supported in the other tool. From the programmers' point of view, this is a major inconvenience. However, it concerns an area that is usually outside the scope of the tasks they perform.

10.Conclusions

Standards for IT in the broadest sense are constantly emerging and evolving. From the study emerges a picture of programmers, however, who value in this field above all not novelty, but stability and simplicity. Operating on text files gives them the freedom to accomplish tasks with the tools of their own choosing. This freedom was emphasized as a very important part of the work. As noted by Raymond (2003) for programmers, standards are one thing, but informal rules that are elements of professional culture are equally important.

In the approach presented by the interviewees, one can see a convergence with the concept called ”Unix philosophy” (McIlroy et al., 1978), and the guidelines contained therein for the way the Unix operating system works. Among the principles set forth therein, relevant in the context of the standards and the described study, it is worth mentioning, among others, the creation of such software whose ”output” will be able to be the ”input” of another – perhaps not yet existing – program. One possible implementation of this idea is specifically using text files described by the respondents, which are a universal way of storing and exchanging information.

It is also worth noting that the fact of defining and publishing a standard is important, but this is not a value in itself. This is because the adoption of the standard in the market and – in the case of software – its correct operation are also important.

In conclusion, standards in the IT industry are crucial for the development of both software and hardware. They are also important in the work of programmers. Of the standards defined in this way, those that are as universal as possible and do not restrict them in their choice of methods and means of accomplishing tasks were particularly highly valued. This confirms the observations of other researchers on this professional group (Jemielniak, 2008; Lin, 2007). In this case, a standard should be understood not only as a formal description of a certain issue – whether by a standardization organization or an industry consortium – but also as a custom and established practice of operation.

Acknowledgements

The publication was financed by the Institute of Social Sciences, SWPS University.

References

1. Arkko, J. (2021). *IETF document statistics (all documents).* Retrieved from: https://www.arkko.com/tools/allstats/, 1.03.2024.
2. Aspray, W. (1997). The Intel 4004 microprocessor: What constituted invention? *IEEE Annals of the History of Computing*, *19*(3), 4-15. https://doi.org/10.1109/85.601727
3. Blind, K., Gauch, S. (2008). Trends in ICT standards: The relationship between European standardisation bodies and standards consortia. *Telecommunications Policy*, *32*(7), 503-513.
4. Blind, K., Gauch, S., Hawkins, R. (2010). How stakeholders view the impacts of international ICT standards. *Telecommunications Policy*, *34*(3), 162-174.
5. Bradley, D. (2011). A Personal History of the IBM PC. *Computer*, *44*(8), 19-25. https://doi.org/10.1109/MC.2011.232
6. Bradner, S. (2003). *IETF structure and Internet standards process*. 78th IETF. Maastricht, Netherlands.
7. Cerri, D., Fuggetta, A. (2007). Open standards, open formats, and open source. *The Journal of Systems and Software*, *80*(11), 1930-1937.
8. Ceruzzi, P.E. (2003). *A history of modern computing* (2nd ed). London, Eng.: MIT Press.
9. Dargan, P.A. (2005). *Open systems and standards for software product development.* Boston: Artech House.
10. David, P.A., Greenstein, S. (1990). The economics of compatibility standards: An introduction to recent research. *Economics of Innovation and New Technology*, *1*, 3-41.
11. Davies, H., Bressan, B. (Eds.) (2010). *A history of international research networking: The people who made it happen*. Weinheim: Wiley-VCH.
12. Duguid, P. (2015). Open standards and the internet age: History, ideology, and networks. *Business History Review*, *89*(2), 357-360. https://doi.org/10.1017 s0007680515000458
13. Geertz, C. (1973). Thick description: Toward an interpretive theory of culture. In: *The interpretation of cultures: Selected essays* (pp. 310-323). New York, NY: Basic Books.
14. Glaser, B.G., Strauss, A.L. (2009). *Odkrywanie teorii ugruntowanej*. Kraków: Nomos.
15. Haigh, T. (2012). The IBM PC: From beige box to industry standard. *Communications of the ACM*, *55*(1), 35-37. https://doi.org/10.1145/2063176.2063188
16. IBM (2006). *The birth of the IBM PC*. Retrieved from: https://www.ibm.com/ibm/history/exhibits/pc25/pc25_birth.html, 1.03.2024.
17. IETF (2021). *Participate in the IETF*. Retrieved from: https://www.ietf.org/about/participate/, 1.03.2024.
18. ISO (2006). *ISO/IEC 26300:2006*. Retrieved from: https://www.iso.org/standard/66363.html, 1.03.2024.

19. ISO (2008). *ISO/IEC 29500-1:2008*. Retrieved from: https://www.iso.org/standard/51463.html, 1.03.2024.
20. Jemielniak, D. (2008). *Praca oparta na wiedzy: praca w przedsiębiorstwach wiedzy na przykładzie organizacji high-tech.* Warszawa: Wydawnictwa Akademickie i Profesjonalne, Akademia Leona Koźmińskiego.
21. Konecki, K. (2000). *Studia z metodologii badań jakościowych: Teoria ugruntowana.* Warszawa: PWN.
22. Konecki, K. (2004). Teoria ugruntowana a kontekst odkrycia. Naturalna historia pewnego badania. In: *W kręgu socjologii interpretatywnej - zastosowanie metod jakościowych.* Świnoujście: ISiP, Uniwersytet Szczeciński.
23. Kostera, M. (1996). *Postmodernizm w zarządzaniu.* Warszawa: PWE.
24. Kostera, M. (2003). *Antropologia organizacji: Metodologia badań terenowych.* Warszawa: PWN.
25. Lin, Y.-W. (2007). Hacker Culture and the FLOSS Innovation. In: K. St. Amant, B. Still (Eds.), *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives.* IGI Global. https://doi.org/10.4018/978-1-59140-999-1
26. McIlroy, M.D., Pinson, E.N., Tague, B.A. (1978). UNIX Time-Sharing System: Foreword. *Bell System Technical Journal*, *57*(6), 1899-1904. https://doi.org/10.1002/j.1538-7305.1978.tb02135.x
27. Muller, M. (2005). Who Owns the Internet? Ownership as a Legal Basis for American Control of the Internet. *Fordham Intellectual Property, Media and Entertainment Law Journal*, *15*(3), 709-748.
28. Murrell, P. (2009). *Introduction to data technologies*. Boca Raton: CRC Press.
29. Raymond, E.S. (1999). *The cathedral & the bazaar: Musings on Linux and open source by an accidental revolutionary.* Cambridge, Mass.: O'Reilly.
30. Raymond, E.S. (2003). *The art of UNIX programming.* Boston: Addison-Wesley.
31. Rogers, E.M. (1962). *Diffusion of innovations.* New York: Free Press.
32. Russell, A.L. (2014). *Open standards and the digital age: History, ideology, and networks.* New York, NY: Cambridge University Press.
33. Updegrove, A. (2007). *Joining Forces: OSDL and the Free Standards Group will become The Linux Foundation*. Retrieved from: https://www.consortiuminfo.org/open-source-open-standards/joining-forces-osdl-and-the-free-standards-group-will-become-the-linux-foundation/, 1.03.2024.
34. van Leeuwen, J., Wiedermann, J. (2001). The Turing Machine Paradigm in Contemporary Computing. In: B. Engquist, W. Schmid (Eds.), *Mathematics Unlimited 2001 and Beyond* (pp. 1139-1155). Berlin/Heidelberg: Springer.
35. Vaughan, J. (2004). *The PC - Personal Computing Comes of Age*. Retrieved from: https://www.ibm.com/ibm/history/ibm100/us/en/icons/personalcomputer/, 1.03.2024.

36. von Neumann, J. (1993). First draft of a report on the EDVAC. *IEEE Annals of the History of Computing*, *15*(4), 27-75. https://doi.org/10.1109/85.238389
37. West, J. (2003). How open is open enough? Melding proprietary and open source platform strategies. *Research Policy*, *32*(7), 1259-1285.

Footnotes

[1] In the following text, differentiation between representatives of the two groups is abandoned; all are referred to as programmers.

[2] Open source software – software whose source code is made publicly available under a license that permits free use, modification and redistribution for any purpose; usually developed in an open collaboration model, that is, by self-organizing teams.

[3] For the purposes of the article, it was considered that free and open source software are the same phenomenon. The differences lie outside of the scope of the paper.

[4] In order for source code to be executed by a computer, it must undergo a process of compilation, i.e. transformation into a form that the machine can ”understand”.

[5] Nowadays, existing ISO standards for encoding are widely used, covering characters outside the English alphabet.

[6] Name comes from Linus Torvalds – the creator of Linux, which is the kernel of the Unix-like operating system.