DOMINIKA BARBARA BONISŁAWSKA*

WYKORZYSTANIE OPEN SOURCE SOFTWARE W DZIAŁANIACH ORGANIZACJI MIĘDZYNARODOWYCH NA PRZYKŁADZIE ONZ I UNII EUROPEJSKIEJ

Abstrakt

W niniejszym artykule scharakteryzowano pojęcie Open Source Software, korzyści i zagrożenie płynące z użycia tego narzędzia, potencjalne płaszczyzny działania oraz jego potencjał biznesowy. Dokonano porównania z terminem Free Source Software w zakresie definicyjnym oraz prawnym. W pracy zaprezentowano także studium przypadku państw, które skutecznie implementują rozwiązania bazujące na otwartym oprogramowaniu, ale także takich, które posiadają duży potencjał w rozwoju tej branży jednak niedostateczne możliwości technologiczne czy wsparcie finansowe ze strony rządowej.

Słowa kluczowe: Open Source Software, Free Source Software, cyfryzacja, nowe technologie, otwarta kultura.

Wstęp

Open source jest wynikiem ewolucji wieloletniej praktyki dostarczania źródeł do oprogramowania. Oprogramowanie tego typu to oprogramowanie z kodem źródłowym, do którego każdy użytkownik ma dostęp, może go sprawdzać, modyfikować i ulepszać. W artykule scharakteryzowano pojęcie Open Source Software, korzyści i zagrożenie płynące z użycia tego narzędzia, potencjalne płaszczyzny działania oraz jego potencjał biznesowy. W celu lepszego zrozumienia terminu dokonano jego porównania z kategorią Free Source Software w zakresie definicyjnym oraz prawnym. Drugą część pracy stanowi z kolei studium przypadku państw, które skutecznie implementują rozwiązania bazujące na otwartym oprogramowaniu. Wskazano również przykłady państw takich, które posiadają duży potencjał w rozwoju tej branży, jednak niedostateczne możliwości technologiczne czy wsparcie finansowe ze strony państwowej.

* Dominika Barbara Bonisławska – absolwentka studiów magisterskich na kierunku Bezpieczeństwo Wewnętrzne. Kontakt e-mail: d.bonislawska@student.uw.edu.pl, dominika.bonislawska2497@gmail.com.

Czym jest Open Source Software?

Pojęcie *open source* zostało stworzone przez Christine Peterson i przyjęte do ogólnego użycia przez Open Source Initiative w 1998 roku. Głównym założeniem autorki było odejście od terminu *free source software*, który przez dwuznaczność językową często był rozumiane przez wzgląd na swoją bezpłatność, a nie na charakteryzującą rozwiązanie otwartość i dostępność. Peterson chciała, żeby pojęcie było zrozumiałe dla osób, które zaczynają się interesować zagadnieniem i dołączają do społeczności OS[1]. Pojęcie *open source*, z języka angielskiego oznacza „otwarte źródło". Celem jego autorów było podkreślenie cech oprogramowania i bazujących na nim produktów, czyli możliwość udostępniania, modyfikowania oraz dostosowywania do potrzeb osoby korzystającej. Wstępnie to podejście było modelem rozpowszechniania kluczowych elementów infrastruktury internetowej (m.in. BIND[2]). Przez dłuższy czas rozwiązania były dostosowywane i dopracowywane przez administratorów, którzy z nich korzystali, chociaż samo rozwiązanie było tworzone przez jedną osobę i/lub w jednym miejscu. Wraz z rozpowszechnieniem się w latach 90. Internetu, Open Source (OS) stało się nie tylko pewnym rodzajem oprogramowania, ale pociągnęło za sobą również rozwój społeczności, która pracując przy projektach jednocześnie rozpowszechniała ideę pracy na ogólnodostępnych materiałach. Rozwój technologiczny sprawił, że osoby lub grupy z różnych miejsc na świecie mogły tworzyć wspólnie projekty. Open Source Software (OSS, open source software, otwarte oprogramowanie) jest to typ oprogramowania, który umożliwia użytkownikowi dostęp do jego kodu źródłowego, czyli sekwencji instrukcji, poleceń, funkcji itp. opisanych przez programistę za pomocą języka programowania. Kod ten, konwertowany jest na język maszynowy, co umożliwia jego odczytanie przez komputer. Równocześnie, użytkownik ma prawo do swobodnej i darmowej modyfikacji kodu według własnych potrzeb, tym samym tworząc pochodną projektu, która może być bezpłatnie lub odpłatnie redystrybuowana[3].

Z czasem, by dopracować definicję OS, stworzono kryteria licencji na zasadach OS (tzw. OSD). Zalicza się do nich:

- *Bezpłatną redystrybucję* – brak ograniczeń dotyczących potencjalnej sprzedaży czy przekazywania kodu źródłowego jako składnika grupowej dystrybucji. Nie powinny być pobierane opłaty licencyjne.
- *Dzieła pochodne/modyfikacje* – licencja musi zezwalać na wszelkie modyfikacje lub projekty, które powstaną pośrednio na tych samych warunkach co przy głównym oprogramowaniu.
- *Integralność kodu źródłowego autora* – licencja może ograniczyć dostęp do zmodyfikowanego kodu źródłowego tylko w sytuacji, kiedy licencja zezwala na dystrybuuję „plików roboczych" z kodem źródłowym w celu modyfikacji w procesie tworzenia.

---

1 C. Peterson, *How I coined the term 'open source*', źródło: https://opensource.com/article/18/2/coining-term-open-source-software [14.01.2021 r.].

2 *Berkeley Internet Name Domain*, rodzaj serwera DNS wykorzystywanego w systemie Linux lub innych systemach typu Unix. Zapewnia stabilną strukturę architektury nazewnictwa domen. Aktualnie wykorzystywana wersja 9, w której starano się uzupełnić luki bezpieczeństwa, źródło: https://www.bind9.net [3.01.2021 r.].

3 A. Amrollahi, M. Khansari, A, Manian, *How Open Source Software Succeeds? A Review of Research on Success of Open Source Software*, „IJICTR" 2014, vol. 6, no. 2, s. 67–68.

Wtedy licencja musi zawierać wyraźną zgodę na rozpowszechnianie oprogramowania opartego na zmodyfikowanym kodzie. Można wymagać w licencji nadawania nowych nazw modyfikacjom/numerów wersji.

- *Zakaz dyskryminacji wobec osób i grup.*
- *Brak dyskryminacji w obszarze działań* – nie można ograniczać wykorzystania programu w jakichkolwiek branżach (medycyna, telekomunikacja, energetyka itp.).
- *Dystrybucja licencji* – licencja dołączona do programu musi dotyczyć wszystkich jego elementów i pochodnych, bez konieczności pozyskiwania dodatkowych dokumentów.
- *Licencja nie może dotyczyć konkretnego produktu* – wszystkie strony, pomimo ewentualnego wyodrębnienia w produkcie, działają na takich samych prawach licencyjnych.
- *Licencja nie może ograniczać innego produktu* – licencja nie może nakładać ograniczeń na inne oprogramowania, które są rozpowszechniane razem z tym oprogramowaniem.
- *Licencja musi być neutralna technologicznie* – licencja nie może narzucać działania na indywidualnej technologii czy stylu interfejsu[4].

Celem OSS jest uczynienie produktu bardziej dostępnym (w tym możliwym do skopiowania), modyfikowalnym i zrozumiałym dla użytkownika[5]. Tym samym programowanie OSS może być wykorzystywane do celów komercyjnych, ponieważ może być sprzedawane. Problemem może pozostać zastrzeżenie projektu, ponieważ, jak zostało wyżej wspomniane, licencja produktu zmodyfikowanego powinna działać na takich samych zasadach, jak licencja oprogramowania pierwotnego.

Projekty Open Source Software

Aby rozpocząć projekt istotne jest zbadanie rynku i tego, co już się na nim znajduje. Wtedy należy zdecydować, czy lepiej wykorzystać istniejące oprogramowanie, czy lepiej jest rozpocząć nowy projekt. W pierwszej możliwości przechodzi się od razu do fazy realizacji, a w drugiej zaczyna się od fazy inicjacji[6].

Istnieje kilka sposobów na rozpoczęcie projektu OSS:

1) Osoba, która dostrzega potrzebę realizacji jakiegoś projektu, zgłasza zamiar i zapotrzebowanie rozwinięcia go publicznie.
2) Programista, który działa na istniejącej bazie kodu, która jest ograniczona, decyduje się udostępnić go publicznie jako pierwszą/bazową wersję programu OS.
3) Kod źródłowy, który został udostępniony publicznie dotyczy dojrzałego projektu.
4) Dobrze rozwinięty i opracowany projekt OS zostaje przejęty i rozbudowany przez zewnętrzną stronę zainteresowaną.

---

4 Open Source Initiative, *The Open Source Definition*, źródło: https://opensource.org/osd [14.01.2021 r.]. Definicja ta oparta jest na Debian Free Software Guidelines.

5 S. Randhawa, *Open Source Software and Libraries*, Centre For Research In Rural And Industrial Development, January 2008.

6 A. Anand, A. Krishna, R. Tiwari, R. Sharma, *Comparative Analysis between Proprietary Software VS. Open-Source Software VS. Free Software*, 5th IEEE International Conference on Parallel, India 2018, s. 144.

Społeczność zaangażowana w projekty jest bardzo różnorodna, jednakże można sprecyzować część cech, które ją charakteryzują. Przede wszystkim w tego rodzaje działań zaangażowane są zazwyczaj osoby prywatne, a nie organizacje. Często deweloperzy mają inną pracę, a projekty realizują w ramach swojego czasu wolnego, traktując to także jako hobby, czy formę samorealizacji. Ze względu na to, że inicjatywy są otwarte, gromadzą ludzi na różnym poziomie zaawansowania. Jednak nie oznacza to, że projekty realizowane są tylko przez osoby niedoświadczone, z niskimi kwalifikacjami. Przez to, że aktualnie na rynku oprogramowań pracuje się zwykle nad bardzo konkretnymi projektami, skupionymi na potrzebach jednego klienta, dla wysoko wykwalifikowanych programistów projekty OS są możliwością poszerzenia swoich horyzontów i wyjścia z pewnych schematów projektowych. Co ważniejsze, to właśnie od tego, jaka grupa deweloperów zgromadzi się do pracy przy projekcie, może zależeć, czy dany projekt osiągnie sukces. Na podstawie działań Apache Software Foundation stwierdzono, że im bardziej różnorodna i aktywna jest społeczność programistów, tym większa szansa na sukces projektu. Poza tym projekty mają zazwyczaj niewielką (do 10 osób) grupę głównych twórców, którzy są odpowiedzialni za rys projektu i stworzenie kodu bazowego. Rola osoby uczestniczącej w danym projekcie zależy od posiadanych kwalifikacji, czasu pracy nad projektem i jej zaangażowania[7]. W przypadku pomyślnego rozwoju projektu, często dołączają nowi uczestnicy, którzy po jakimś czasie mogą się wycofać z działania lub przejść do innych inicjatyw. Nowi deweloperzy mogą wnieść świeże spojrzenie, czy wyłapać błędy, które do tej pory nie zostały zauważone. To one mogą stanowić również źródło nowej motywacji dla zespołu[8].

Zastanawiające może być to, dlaczego ludzie wykazują zainteresowanie długoterminową, często darmową pracą. W swoim artykule Alexander Hars i Shaosong Ou precyzują dwie główne płaszczyzny motywacyjne dla uczestników projektów OS:

- wewnętrzne, które dotyczą kwestii samorealizacji oraz
- zewnętrzne, które są zależne od środowiska, często dotyczą potencjalnych korzyści materialnych lub rozpoznawalności wynikającej z uczestnictwa[9].

W przeprowadzonym przez autorów badaniu okazało się, że różnice motywacyjne widoczne są na poziomie dwóch grup badanych. Studenci i osoby zajmujące się tym hobbystycznie kierują się głównie swoją wewnętrzną potrzebą (81,8%) oraz altruistycznymi pobudkami (24%), podczas gdy programiści kontraktowi – osoby w ten sposób zarabiające – motywowani są głównie przez swoją wewnętrzną potrzebę, ale też ze względu na ekonomiczne korzyści płynące z ich działania na przyszłość (odpowiednio 68,2% oraz 53,2%). W związku z tym można wysnuć wnioski, że rynek ze względu na rodzaje motywacji deweloperów jest bardzo zróżnicowany, dzięki czemu może stać się bezpośrednią konkurencją dla tradycyjnego rynku software'owego[10].

Oprócz motywacji opisanych z perspektywy jednostki, można też przeanalizować, w jakich obszarach życia i na jaką skalę może pojawiać się motywacja do pracy przy pro-

7 S. Weerawarana, J. Weeratunga, *Open Source in Developing Countries*, SIDA, 2004, s. 16.

8 Tamże, s. 16.

9 A. Hars, S. Ou, *Work for free? Motivations for participating in Open-source projects*, „International Journal of Electronic Commerce" 2002, vol. 6, no. 3, s. 26–30.

10 Tamże, s. 34–35.

jektach OS. W raporcie SIDA, opracowanym na podstawie badań Josepha Fellera i Briana Fitzgeralda, przytoczono trzy główne płaszczyzny: technologiczną, ekonomiczną i społeczno-polityczną. Każdy z obszarów dodatkowo przebadano z perspektywy indywidualnej (*micro*), jak i organizacji/środowiska (*macro*)[11].

Przykłady motywacji przedstawia poniższa tabela.

Tabela 1. Rodzaje motywacji wg Josepha Fellera i Briana Fitzgeralda

Techniczna micro	– chęć realizacji „potrzeby technicznej", którą ktoś zaobserwował, – praca z najnowocześniejszą technologią.
Techniczna macro	– możliwość podziału czaso- i pracochłonnych projektów między innych programistów, – promocja innowacji, – możliwość pracy ze społecznością nad badaniami i rozwojem, – przejrzystość działań dzięki posiadaniu otwartego kodu źródłowego.
Ekonomiczna micro	– uzyskanie korzyści na przyszłość zawodową, – rozwój i poprawa umiejętności kodowania, – możliwość lepszych zarobków, – niskie koszty alternatywne.
Ekonomiczna macro	– obniżenie kosztów technicznych związanych z wykorzystaniem otwartego oprogramowania, – możliwość działania na skalę międzynarodową, – wytworzenie dochodów pośrednich wynikających z produktów, usług etc.
Społeczno-polityczna micro	– altruizm, – poczucie przynależności do społeczności, – poczucie realizacji oraz bycia docenionym, po tym jak kod jest wykorzystywany przez innych.
Społeczno-polityczna macro	– kreowanie nowych trendów/modelu pracy, – ideologia (w przypadku społeczności Free Source Software), – pomoc w zmniejszaniu „przepaści cyfrowej".

Źródło: opracowanie własne za: J. Feller, B. Fitzgerald, *Understanding Open Source Software Development*, Addison-Wesley, London 2002.

Badanie dodatkowo akcentuje fakt, że osoby, które decydują się uczestniczyć w projektach, nie muszą kierować się jedynie dobrą wolą czy altruizmem, ale również chęcią realnego rozwoju w perspektywie zdobytych umiejętności, jak i potencjalnych szans ekonomicznych.

Historia OSS i jego uwarunkowania prawne

Od lat 60. do 80. XX wieku, rynek sprzętu technologicznego był skupiony przede wszystkim na sprzedaży i dostarczeniu sprzętu komputerowego (hardware), który miał specjalnie dostosowywane oprogramowanie. W tamtym czasie specjaliści IT rozpoczęli

[11] S. Weerawarana, J. Weeratunga, dz. cyt., s. 17.

badania nad bardziej uniwersalnym oprogramowaniem, które pasowałoby do wielu urządzeń. Jako rezultat tych prac w 1969 roku, powstał Unix. Dostęp do niego dla jednostkowych użytkowników był drogi, przez co jego wykorzystanie było marginalne. Dużo łatwiej dostępny był na uniwersytetach, które płaciły tylko część bazowej ceny. Dzięki temu eksperci rozpoczęli swoje prace nad rozwojem technologii internetowych. Powszechną praktyką stało się wtedy dzielenie się kodem źródłowym oprogramowania. Autorzy Unixa zablokowali jednak tę możliwość prawnie, przez co inne podmioty na rynku (m.in. IBM) rozpoczęły pracę nad swoimi wersjami oprogramowania, podobnego do Unix. Działania te przebiegały w większych grupach, które wspólnie, symultanicznie pracowały nad kodem. Istotnym momentem było pojawienie się na rynku produktu stworzonego przez Richarda Stallmana, który w latach 80. wprowadził GNU (GNU's Not Unix) oraz przedstawił filozofię Free Source Software, co zostanie szerzej opisane w dalszej części.

Lata 90. sprzyjały rozwojowi Internetu oraz *World Wide Web*[12], przez co na popularności zyskiwały rozwiązania oparte na otwartym oprogramowaniu. Tym samym pojawiła się potrzeba komercjalizacji powstałych rozwiązań czy inicjatyw. Dzięki temu korzyści finansowe, ale też rozwojowe, miały być osiągalne zarówno dla społeczności OS, jak i potencjalnych konsumentów. W 1998 roku odbyła się sesja strategiczna w Palo Alto, w Kalifornii, podczas której grupa specjalistów: Todd Anderson, Larry Augustin, John Hall, Sam Ockman, Christine Peterson i Eric S. Raymond, miała otrzymać dostęp do kodu źródłowego firmy Netscape[13]. Było to spowodowane pojawieniem się na rynku publikacji Erica Raymonda, *The Cathedral and The Bazaar*, w której wyjaśniono wagę i użyteczność korzystania z OSS oraz szanse dla przedsiębiorców, które się za tym kryły. Przekazanie przez firmę kodu, było jednym z pierwszych działań na tak dużą skalę w celach komercyjnych. Nikt nie miał wtedy pewności czy próba pracy nad tym produktem w trybie open source się powiedzie, jednakże był to krok milowy zarówno dla samego ruchu OS, jak i rynku, na którym wciąż nie było pewności co do tego typu rozwiązań. Sam projekt zakończył się sukcesem, który nie tylko przekreślił rolę Microsoftu na rynku jako monopolistycznego dostawcy przeglądarek internetowych[14], ale podkreślił wagę tego, jakie finansowe oraz edukacyjne możliwości niesie za sobą ogólnodostępna, współdzielona, jawna wiedza[15]. Powszechnie sesja ta uważana jest za moment powstania pojęcia open source. Jak wspomniano wcześniej, ostatecznie termin ten przyjęty został w 1998 roku przez Open Source Initiative.

Do zrozumienia zróżnicowania, jakie powstało w tamtym czasie w dyskursie między *proprietary software* (oprogramowanie własnościowe) a OSS, najlepiej przysłużyło się porównanie Raymonda w jego wspomnianej już publikacji. Tytuł pracy prezentuje dwa symboliczne koncepty, dotyczące powstawania i działania obu rodzajów oprogramowania.

---

12 W tłumaczeniu: *Ogólnoświatowa sieć.* Ogólnodostępna baza danych, jedna z usług internetowych.

13 Netscape była firmą świadczącą usługi komputerowe, znaną z jednej z najbardziej popularnych wyszukiwarek sieciowych w latach 90. – Navigator. Udostępnienie podczas tej konferencji kodu do ogólnego dostępu było przełomowym krokiem, ponieważ rozpoczęło OS projekt Mozilla, który ostatecznie pozwolił na wykształcenie przeglądarki Firefox, źródło: https://www.oreilly.com/openbook/opensources/book/netrev.html [23.08.2021 r.].

14 E.S. Raymond, *The Cathedral and The Bazaar Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly Media, Newton 1999, s. 61–62.

15 *History of the OSI*, źródło: https://opensource.org/history [20.01.2021 r.].

Zgodnie z nimi oprogramowanie własnościowe swoją strukturą przypomina budynek/projekt katedry, podczas gdy otwarte oprogramowanie, w tym sposób komunikacji tej społeczności, kojarzy się z orientalnym bazarem[16].

Różnice wynikające z tej teorii prezentuje poniższa tabela.

Tabela 2. Różnice między oprogramowaniem otwartym a własnościowym na podstawie The Cathedral and The Bazaar

	Proprietary Software (*Katedra*)	**Open Source Software (*Bazar*)**
Planowanie	Odgórnie całościowo	Etapowo
Cel	Zrealizowanie kontraktu	Rozwiązanie problemu
Dyscyplina pracy	Silna	Słaba
Finansowanie/zasoby	Znane	Nieznane
Szybkość reagowania	Wolne	Sprawne
Postęp prac	Ukryty	Jawny
Sposób współpracy	Osobiście	Online/Zdalne

Źródło: opracowanie własne za: E.S. Raymond, *The Cathedral and The Bazaar Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly 1999, s. 21–63.

Podsumowując rozważania autora w tej kwestii, można stwierdzić, że optuje on za opcją korzystania z otwartego oprogramowania. W porównaniu do oprogramowania zamkniętego, jego korzyści przejawiają się na wielu płaszczyznach. Jednocześnie podkreśla fakt, że społeczność OS sama dla siebie pracuje na swoje rezultaty, a także że projekty konkurują między sobą, co napędza działanie i motywację do doskonalenia całego systemu[17].

Free Source Software a Open Source Software

Prowadząc rozważania na temat Open Source Software należy zaznaczyć różnice między tym pojęciem, a terminem Free Source Software. Są one czasem stosowane zamiennie w literaturze, ale dotyczą dwóch różnych materii.

Termin OS został stworzony w celu określenia typu oprogramowania i tworzenia programów komputerowych. Jednak dziś pojęcie to jest już dużo szersze – powstała pewna *ścieżka*/*droga open source*. Mogą to być zarówno produkty, jak i projekty, które korzystają z otwartych kodów źródłowych. Opierają się na zasadach otwartej wymiany, współpracy, hierarchii opartej na merytokracji, sprawnego prototypowania, a przede wszystkim rozwoju skupionym na społeczności[18].

16 E.S. Raymond, dz. cyt., s. 19.

17 Tamże, s. 58.

18 S. Randhawa, dz. cyt., s. 369–370.

Z kolei Free Source Software to rodzaj oprogramowania, które w swobodny sposób może być używane, modyfikowane oraz rozpowszechniane, ale pod warunkiem, że każda dystrybucja będzie zawierała oryginalne warunki wolnego użycia, modyfikacji oraz dystrybucji (tzw. *copyleft*[19]). Cały projekt i jego koncept został stworzony przez Richarda Stallmana, który w 1983 roku pracował nad alternatywą dla systemu operacyjnego Unix i w tym samym czasie wypracował specjalną licencję o nazwie GNU (Gnu's Not Unix). W ramach wsparcia swojego pomysłu, kilka lat później, założył fundację Free Software Foundation (FSF), która wspiera członków szczególnie w aspekcie prawnym przy przekazywaniu praw autorskich.

Głównym celem GNU jest 100% *free software* – otwarte, dostępne oprogramowanie, gdzie użytkownicy mogą swobodnie powielać, zmieniać, sprawdzać czy dostosowywać produkt do swoich potrzeb. Słowo *free* w rozumieniu organizacji nie oznacza ceny, ale tego, że użytkownik ma wolność. Używa się czasem zamiennie słowa *libre* (software), dla podkreślenia, że nie chodzi o aspekt finansowy. Stallman zakłada, że jeśli użytkownik nie ma kontroli nad oprogramowaniem, to wtedy oprogramowanie ma kontrolę nad nim[20]. Tym samym, wyróżnia cztery rodzaje wolności, których występowanie świadczy o tym, że użytkownicy mają do czynienia z FS:

1) *Wolność 0* – swobodne uruchamianie programu, w dowolnym celu.
2) *Wolność 1* – możliwość swobodnej analizy tego, w jaki sposób działa program oraz dostosowywania go do własnych potrzeb. Warunkiem koniecznym jest dostęp do kodu źródłowego.
3) *Wolność 2* – możliwość rozpowszechniania kopii, aby pomagać innym.
4) *Wolność 3* – swoboda w ulepszaniu programu oraz udostępnianiu swoich poprawek, dzięki czemu społeczność również może z nich skorzystać. Warunkiem koniecznym jest ponownie dostęp do kodu źródłowego[21].

Autor zakłada, że jeśli program nie zawiera którejś z tych wolności, to z założenia jest nieetyczny. Wtedy należy ono prawdopodobnie do grupy *proprietary software*, czyli programów, które zostały stworzone przez osobę fizyczną lub organizację/firmę i pozostają ich własnością. Oznacza to, że nie ma wolnego dostępu do kodu źródłowego, który umożliwiałby swobodne modyfikacje czy pracę na programie. Udostępnienie kodu do domeny publicznej przez właściciela sprawia, że plik nie jest chroniony prawami autorskimi i może być powszechnie wykorzystywany[22].

Razem z pojawieniem się na rynku tych dwóch form oprogramowania, pojawiła się potrzeba dostosowania sytuacji prawnej związanej z dostępem do oprogramowania typu OS. Razem z powstaniem Free Source Software (FSS) stworzony został rodzaj licencji zwany GNU General Public License (GPL). Dokument ten działał zgodnie z zasadą *copyleft*, która, pomimo wprowadzania modyfikacji w plikach, zobowiązywała do podlegania

---

19 Platforma GNU Operating System, *What is Copyleft?*, źródło: https://www.gnu.org/licenses/copyleft.html [15.01.2021 r.].

20 Platforma GNU Operating System, *The GNU Operating System*, źródło: https://www.gnu.org/gnu/gnu.html [15.01.2021 r.].

21 Platforma GNU Operating System, *What is free software?*, źródło: https://www.gnu.org/philosophy/free-sw.html [15.01.2021 r.].

22 A. Anand, A. Krishna, R. Tiwari, R. Sharma, dz. cyt., s. 144.

pod tę licencję. Tym samym tworzyło to *wirusowy efekt*, który uniemożliwiał wykorzystywanie stworzonych produktów w celach komercyjnych, gdzie zyski można byłoby osiągać na podstawie umów licencyjnych. W odpowiedzi na to, Open Source Initiative (OSI) wypracowując wyżej wspomnianą OSD (która składa się na licencję OS) wyznaczyło wskazówki dla licencji OSS, które są jednocześnie znakiem towarowym dla wypracowanych projektów i rozróżniają je od wytworów GPL. Chcąc uzyskać akceptację OSI licencja powinna spełniać określone kryteria formalne i czasowe. Dokument nie może się powielać lub przypominać wcześniej zatwierdzonych już licencji, musi być zgodny z OSD, a całość powinna mieć dołączone prawne uzasadnienie. Komisja OSI ma 60 dni na przeanalizowanie zgłoszenia, następnie po odesłaniu dokumentu z poprawkami, OSI otrzymuje kolejne 30 dni na zatwierdzenie nowej licencji i zaktualizowanie listy[23]. Na stronie internetowej Open Source Initiative dostępna jest lista licencji, które zostały przez komitet do tej pory zaakceptowane i zatwierdzone jako zgodne z definicją OS[24].

Podsumowując, pomimo tego, że podstawa działania obydwu podejść była podobna, to główna różnica polegała na tym, że założyciele OSI chcieli podkreślić praktyczność korzystania z otwartego oprogramowania, podczas gdy twórcy FSS zawierali w swoim podejściu kwestie etyczne bezwzględnego prawa użytkowników do dostępu do kodu źródłowego (z możliwością edytowania go do własnych potrzeb). Innymi słowy, aktualnie pojęcia wykorzystywane są zamiennie nieświadomie.

Korzyści, potencjalne ryzyko, model skuteczności systemów informatycznych

Wprowadzanie otwartego oprogramowania do działań na poziomie lokalnym, krajowym czy międzynarodowym wiąże się z wyzwaniami, które wynikają ze szczególnego charakteru produktów oraz sposobu pracy nad nimi. Dodatkowo można do tego dołączyć specyficzną sytuację prawną wykreowanych programów. Jednakże przez lata zauważono, że działania, które wykorzystywały opcję OSS, kończyły się sukcesem. Wyszczególnić tu można inicjatywy *peer-produced*[25], jak na przykład Wikipedia. W związku z tym warto zastanowić się jakie korzyści i zagrożenia mogą płynąć z implementacji takiego oprogramowania.

Bazując na badaniu zorganizowanym przez Lorraine Morgan i Patricka Finnegana można wyszczególnić kilka zalet i wad płynących z wykorzystania OSS. Skupiono się w nim, w głównej mierze, na wykorzystaniu tego typu oprogramowania w administracji publicznej, dzieląc na dwie płaszczyzny: techniczną i biznesową[26].

---

23 Platforma Open Source Initiative, *The License Review Process*, źródło: https://opensource.org/approval [15.01.2021 r.].

24 Platforma Open Source Initiative, *Licenses by Name*, źródło: https://opensource.org/licenses/alphabetical [15.01.2021 r.].

25 Produkty wytworzone rówieśniczo, czyli samoorganizująca się społeczność gromadzi się w celu jej stworzenia.

26 L. Morgan, P. Finnegan, *Benefits and Drawbacks of Open Source Software: An Exploratory Study of Secondary Software Firms*, Limerick, Ireland 2007, s. 308.

Tabela 3. Zalety i wady OSS z perspektywy technicznej na podstawie badanie L. Morgan i P. Finnegana

	Zalety	Wady
Techniczne	Rzetelność/Niezawodność; Bezpieczeństwo (częściowo); Jakość; Elastyczność użycia; Dostępność programistów i grup testujących produkt; Wydajność; Kompatybilność; Harmonizacja prawa.	Brak ekspertyz; Słaba jakościowo dokumentacja; Bezpieczeństwo (częściowo); Mniejsza funkcjonalność; Rozprzestrzenianie różnych interfejsów; Brak usystematyzowanego procesu tworzenia (myślowego).
Biznesowe	Niskie koszty; Zaangażowanie do bycia innowacyjnym; Elastyczność wynikająca z charakteru licencji; Wzrost współpracy; Dodatkowa funkcjonalność biznesu; Tworzenie standardów na rynku; Możliwość wyboru, ucieczka od „schematów" wykorzystywanych na rynku.	Brak właściciela; Brak wsparcia technicznego; Brak kontroli nad dostępem do kodu źródłowego; Niewystarczający marketing; Trudność ze znalezieniem odpowiednich pracowników.

Źródło: opracowanie własne za: L. Morgan, P. Finnegan, *Benefits and Drawbacks of Open Source Software: An Exploratory Study of Secondary Software Firms*, Limerick, Ireland 2007, s. 310.

Biorąc pod uwagę aspekty techniczne najważniejszą zaletą wydaje się być niezawodność, czyli fakt, że produkty są szeroko dostępne a poziom uniwersalności wielu projektów jest wysoki. Jakość jest tu rozumiana jako możliwość poprawiania produktu i dopasowywania go do potrzeb użytkowników. OSS umożliwia wykorzystanie innowacyjnych sposobów działania, dając równocześnie możliwość zdobycia aktualnej wiedzy i rozwoju umiejętności osobom, które się tym zajmują. Ponadto implementacja OSS na większą skalę wpływa pozytywnie na kompatybilność i lepszą interoperacyjność[27].

Z perspektywy biznesowej najbardziej znaczącym czynnikiem przemawiającym za wykorzystaniem OSS jest potencjalna redukcja kosztów (brak opłat licencyjnych, mniejsze koszty systemów antywirusowych etc.). Korzystanie z otwartego oprogramowania przez coraz większą liczbę przedsiębiorstw naturalnie powoduje wytworzenie na rynku oprogramowań nowych standardów oraz trendów, ale także pozwala pracownikom na rozwinięcie ich umiejętności oraz zaproponowanie innowacyjnych podejść do problemów[28]. Zachęca jednocześnie do współpracy, która dzięki temu, jaki charakter ma OS, jest aterytorialna. Nawet jeśli projekt jest realizowany w celu rozwiązania jakiegoś wyzwania, a autorzy teoretycznie skupiają się na swoich osobistych korzyściach, to i tak rozwiązanie przyniesie pozytywne rezultaty dla ogółu. Coraz częściej można się również spotkać z założeniem, że klienci liczą na to, że rozwiązania zaproponowane będą opierały się na zasadach otwartej licencji[29].

[27] Tamże, s. 309.

[28] Tamże, s. 309–310.

[29] S. Weber, *The Political Economy of Open Source Software*, „BRIE Working Paper", 15 June 2000, s. 3.

Dyskusyjna pozostaje kwestia bezpieczeństwa, ponieważ podczas rozbudowanej współpracy istnieje szansa, że luki bezpieczeństwa w kodzie zostaną zauważone, a tym samym będzie on lepiej rozwinięty i wzmocniony[30]. Jednocześnie praca nad kodem pozwala rozwinąć poczucie odpowiedzialności oraz świadomości na temat kwestii bezpieczeństwa[31].

Chociaż istnieje dużo czynników przemawiających za tym, że produkty OSS są bezpiecznym rozwiązaniem dla użytkowników, to wciąż można wyszczególnić kilka aspektów, które wpływają na podatność oprogramowania na zagrożenia. W raporcie *Open Source Security and Risk Analysis 2020*, przygotowanym przez firmę Synopsys, znaleźć można informację, że 75% kodów źródłowych z firm, które zostały przez nich przebadane, posiada w swoich strukturach słabości. Zaleca się, żeby obserwować pojawiające się na rynku aktualizacje, ponieważ wersje kodu są wtedy dopracowywane, a tym samym bezpieczniejsze[32]. Z drugiej strony, otwarty dostęp do kodu umożliwia hakerom wynajdowanie błędów i słabości, na których mogą przeprowadzić swoje działania. Informacje wrażliwe, jak dane osobiste użytkowników, mogą wtedy zostać wykradzione. Jednakże w powyższym raporcie zaznaczono również, że nie jest możliwe zidentyfikowanie i naprawienie każdego problemu, który pojawia się w kodzie. Tym samym aktualna polityka wprowadzania poprawek do kodu powinna ulec zmianie – komercyjni sprzedawcy mogą skupić się na rozpowszechnianiu aktualizacji oraz uświadamianiu o kwestiach bezpieczeństwa, jednak zmiany w kodzie powinny zachodzić w jego bazowej wersji[33].

Powyższe rozważania mogą sprowokować pytanie, na ile realizowane projekty mają szansę odnieść sukces lub czy istnieje sposób monitorowania tego, jakich systemów informatycznych potrzebują społeczeństwa w różnych państwach. Tworzenie projektów systemów informatycznych związane jest z różnorodnymi problemami społecznymi, behawioralnymi czy technicznymi, dlatego analiza czy projekt odniesie/odniósł sukces jest utrudniona[34]. Nie ma pewności, czy pomimo licznych korzyści uda się inicjatywie przetrwać i rozwinąć na rynku. Bazując na badaniach oraz obserwacjach, William Delone oraz Ephriama McLean wypracowali w roku 1992 model (zaktualizowany dziesięć lat później) dotyczący tego, jakie systemy informatyczne mają szansę osiągnąć sukces.

Według nich, co można zaobserwować na grafie poniżej, jakość dzieła należy rozpatrywać na trzech płaszczyznach – jakość informacji, jakość systemów oraz jakość usług. Każda z nich powinna być badana osobno, ponieważ indywidualnie (ale też wspólnie) mają wpływ na poziom satysfakcji użytkowników. Pojęcia *intention to use* oraz *use* są szerokim spojrzeniem na to jakie jest podejście do danego systemu – czy użycie jest wolontaryjne czy obligatoryjne, efektywne czy nieefektywne. Badacze chcieli pozostawić możliwość brania pod uwagę kwestii behawioralnych użytkowników, ale nie ma takiego obowiązku ze względu na to, że czynniki te są często losowe i niejednoznaczne. Używanie systemu (*use*)

30 P. Banerjee, *On the security of Open Source Software*, International Journal of Advanced Research, November 2017, s. 1339.

31 Tamże, s. 1345.

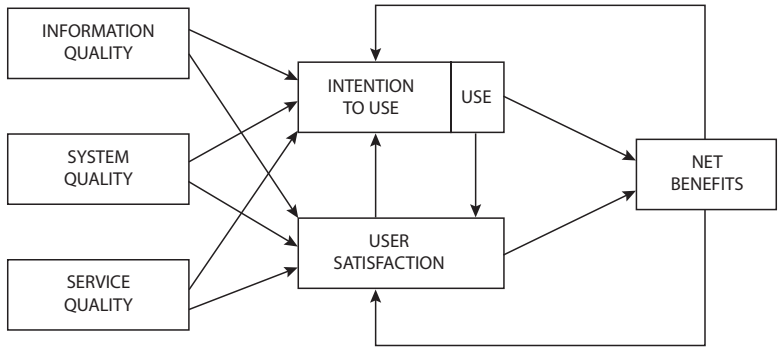
32 Synopsys, *Open Source Security and Risk Analysis 2020*, s. 6, źródło: https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html [6.03.2021 r.].

33 Tamże, s. 14.

34 A. Amrollahi, M. Khansari, A, Manian, dz. cyt., s. 67.

może prowadzić do zwiększenia satysfakcji użytkownika, która jednocześnie doprowadzi do większej chęci użycia (*intention to use*) oraz samego użycia. *Net benefits* to korzyści jakie osiąga konsument na każdej płaszczyźnie handlu elektronicznego. Innymi słowy, na ile jego działania w przestrzeni online są płynne, satysfakcjonujące, bezproblemowe. Aspekt sprzedaży online jest konieczny dla zachowania równowagi między pozytywnym a negatywnym wpływem handlu elektronicznego na klientów, rynki, organizacje, a nawet całe społeczeństwa. Poza tym działania online pozwalają w pełni wykorzystywać potencjał systemów[35].

Grafika 1. Zaktualizowany D&M IS Success Model



Źródło: W. Delone, E. McLean, *The DeLone and McLean Model of Information Systems Success: A Ten-Year Update*, Journal of Management Information Systems, April 2003, s. 24.

Podsumowując, wyżej zaprezentowany model pokazuje, jak bardzo powiązane są ze sobą czynniki, które na różnych płaszczyznach wpływają na potencjalny sukces systemów informatycznych. Według autorów należy pamiętać o dostosowaniu wszelkich danych do potrzeb konkretnego badania empirycznego. Co najważniejsze, jest to jedna z opcji, która może zostać wykorzystana nie tylko do badania poziomu skuteczności mniejszych, lokalnych projektów, ale także przedsięwzięć na skalę co najmniej krajową, przez co może stać się sposobem do monitorowania potrzeb i sukcesów przedsięwzięć podejmowanych w różnych regionach na świecie[36].

Formy implementacji OS w państwach rozwijających się

Biznesowy potencjał OSS

Jedną z korzyści płynących z wykorzystania OSS i produktów z nim związanych jest jego rosnący potencjał biznesowy. Coraz więcej firm z branży IT wykorzystuje rozwiązania oparte na otwartej licencji. Najbardziej popularne są dwie opcje zaangażowania firm – jako dystry-

35 Tamże, s. 23–25.

36 Tamże, s. 27–28.

butorzy i sprzedawcy produktów albo jako firmy oferujące wsparcie techniczne w zakresie usług OS. Działania mogą być prowadzone na dwóch głównych rynkach – lokalnym oraz globalnym. Pierwszy, ze względu na szerzący się proces globalizacji, jest za mało konkurencyjny w porównaniu do projektów międzynarodowych. W związku z tym zakłada się, że najlepiej, jeśli projekty tworzone są w celu ulepszenia i podniesienia wartości produktów międzynarodowych. W przypadku modelu skupionego na eksporcie usług i produktów OS niezbędnym pozostaje posiadanie odpowiednio rozwiniętej struktury informatycznej oraz wykwalifikowanej kadry. Dzięki temu państwo może świadczyć usługi *outsourcingu* dla innych, a tym samym staje się niezależne w tym obszarze usług technologicznych[37].

Zauważając biznesowe możliwości wykorzystania OSS, w raporcie zrealizowanym przez SIDA zaprezentowano, jak wpływa to bezpośrednio na państwa rozwijające się i jakie możliwości im to daje. W dokumencie zauważa się, że główne korzyści płynące dla państw, które zgodziły się wprowadzić zmiany na korzyść wykorzystania OSS, to przede wszystkim redukcja kosztów, autonomia (brak zależności od zewnętrznych firm oferujących oprogramowania), wzrost produktywności oraz poprawa jakości obowiązującego prawa dotyczącego własności intelektualnej.

Według danych zawartych w tym raporcie implementacja rozwiązań OS ma istotne znaczenie we wstępowaniu na ścieżkę rozwoju technologicznego, w kontekście postępującego procesu digitalizacji na świecie. Wiąże się to bezpośrednio z tym, czego oczekują użytkownicy. Firmy publiczne i prywatne mierzą się z rosnącym popytem na usługi, które powinny stać się prostsze, szybsze, ich jakość wyższa, a cena wykorzystywanych technologii niższa. W związku z tym przedsiębiorstwa, w celu odpowiedzenia na popyt, decydują się na wykorzystanie OS na poziomie biznesowym[38].

Kolejnym znaczącym aspektem wdrażania rozwiązań OSS jest kwestia bezpieczeństwa oraz autonomiczności. Państwa, dzięki osiągnięciu niezależności technologicznej od dużych firm sprzedających swoje oprogramowania, mogą reagować lokalnie na problemy czy błędy pojawiające się w ramach korzystania z oprogramowania. Innymi słowy nie muszą powierzać kwestii bezpieczeństwa firmom zewnętrznym, których biura wsparcia mogą być zlokalizowane w zupełnie innej części globu, co negatywnie wpływa na tempo reagowania na sytuacje kryzysowe. Przykładem może być tu rozwijająca się współpraca między Koreą Południową, Chinami oraz Japonią, które będą się starały dostosować oprogramowanie Linux do własnych potrzeb rynkowych[39].

Poza tym wykorzystywanie produktów OS na poziomie krajowym sprawia, że krajowe systemy stają się bardziej transparentne, również dzięki temu, że użytkownicy mają do nich darmowy dostęp. Wykorzystanie tego typu oprogramowań, nie tylko stymuluje rozwój kompetencji technologicznych na rynku, ale i daje priorytet szeroko pojętej modernizacji. A w sytuacji, kiedy pojawi się nowa i lepsza wersja wykorzystywanego produktu, dużo łatwiej dokonać transferu danych oraz uaktualnić system.

37 S. Weerawarana, J. Weeratunga, dz. cyt., s. 23–26.

38 Tamże, s. 28.

39 Tamże, s. 29.

Sposoby implementacji rozwiązań OS

Zakładając, że wprowadzenie rozwiązań OS może przynieść korzyści dla państw rozwijających się, należy zastanowić się, na jakich płaszczyznach należy podjąć działania, by zapewnić wydajność i efektywność w tym zakresie. Steven Weber w swojej publikacji *Open Source Software in Developing Economies* prezentuje, w jakich obszarach rządy mogą działać, żeby usprawnić proces wprowadzania zmian:

- Działania na poziomie formalnym (tworzenie prawa, strategii w tym obszarze oraz planów) i nieformalnym (pozostawienie przestrzeni dla OSS, żeby mogło ewoluować).
- Zaangażowanie na różnych poziomach: lokalne, krajowe i/lub regionalne możliwości współpracy.
- Tryby rozwoju: publiczny (wprowadzanie rozwiązań typu OS) oraz prywatny (wprowadzanie rozwiązań oraz odpowiadanie na różnorodne potrzeby rozbudowanej grupy użytkowników)[40].

Kolejną kwestią, od której może zależeć potencjalny sukces wprowadzania otwartego oprogramowania na poziomie państwowym, są obszary, które powinny działać wystarczająco sprawnie, żeby OSS zaczął przynosić korzyści. W tym zakresie Weber wymienia:

- Politykę technologii informacyjnych (polityka IT).
- Rzecznictwo (ang. *advocacy*) oraz edukacja.
- Potencjał do tworzenia lokalnego przemysłu oprogramowań.
- Pozycję lokalnego przemysłu na rynku globalnym.
- E-administrację[41].

Pierwsza sfera działań zakłada stworzenie zrównoważonego, wolnego od dyskryminacji i otwartego środowiska, które będzie sprzyjało współpracy. Związana jest z tym także kwestia własności intelektualnej, która nie tylko nie pozwoli na nadużycia, ale będzie kreowała postawę odpowiedzialności oraz szacunku do tego typu własności w społeczeństwie. Dodatkowo brany jest tu pod uwagę aspekt darmowego dostępu do Internetu, który jest niezbędny w pracy nad tego typu projektami, ale też Internetu, który nie jest odgórnie kontrolowany czy ograniczany przez władzę. W ramach rozwoju polityki IT znaczące byłoby włączenie w cały proces również sektora prywatnego oraz zachęcenie go do korzystania z tego typu rozwiązań[42].

W przypadku rzecznictwa i edukacji zaproponowane zostało między innymi stworzenie grupy eksperckiej w zakresie działań OS, która mogłaby oferować wsparcie merytoryczne. Oprócz tego według autora należałoby rozpocząć promocję tego typu rozwiązań, na przykład poprzez udostępnianie komputerów działających na takich oprogramowaniach lub stworzenie stron, które informowałyby o aktualnie tworzonych czy rozwijanych projektach. Z perspektywy edukacji, umiejętność pracy z projektami typu OS powinna być jedną z tych, które mogą zostać zdobyte w ramach zajęć szkolnych/uniwersyteckich. Co więcej, trzeba zaoferować obywatelom możliwość uczestnictwa w szkoleniach poszerzających ich kompetencje w tym zakresie. Dotyczy to również edukowania programistów w języku angielskim[43].

40 S. Weber, *Open Source Software in Developing Economies*, University of California, Berkeley 2003.

41 S. Weerawarana, J. Weeratunga, dz. cyt., s. 37–46.

42 Tamże, s. 35, 37–38.

43 Tamże, s. 40–41.

Jak powszechnie wiadomo język angielski jest podstawowym językiem wykorzystywanym w programowaniu. Jeśli jednak kraj chciałby bardziej odpowiedzieć na potrzeby społeczeństwa, to w ramach rozwoju potencjału lokalnego sugerowane jest skupienie się na stworzeniu odpowiedniego centrum wsparcia technicznego udzielającego pomocy w języku ojczystym. Takie placówki mogą działać na terytorium kraju, ale także poza nim. Z drugiej strony duże znaczenie ma to, jak kraj prezentuje swoje możliwości na arenie międzynarodowej. Dlatego też polecane jest uczestnictwo w inicjatywach na skalę większą niż krajowa oraz otwartość na możliwość współpracy. Nie można tu pominąć również kwestii posiadania zaplecza specjalistów, którzy są konkurencyjni na rynku[44].

Ostatnią płaszczyzną jest kwesta e-administracji. W celu zrównoważonego rozwoju tej materii niezbędne jest stworzenie takich ram prawnych, w których produkty OS miałyby równą szansę i możliwość zaistnienia na rynku równolegle do *proprietary software*[45].

Wysiłki na rzecz rozwoju wyżej wymienionych płaszczyzn będą miały pozytywny wpływ nie tylko na wzrost poziomu digitalizacji w państwie, ale także na zaangażowanie społeczeństwa i zrozumienie przez niego zachodzących zmian technologicznych. Jednocześnie, jeśli państwa planują wprowadzanie takich innowacji, niezbędnym jest odpowiednie przeanalizowanie tego, na jakim poziomie rozwoju znajdują się wpływające na to elementy. Pozwala to na dopasowanie strategii do potrzeb lokalnych, a tym samym sprecyzowanie, jakie podmioty i w jaki sposób mogą być włączone do tych działań. Głównymi komponentami, na których należy się skoncentrować, jest polityka IT oraz infrastruktura IT razem z umiejętnościami w tej branży.

Tabela 4. Zależność między Polityką IT a infrastrukturą i umiejętnościami IT w zależności od poziomu rozwoju na podstawie raportu SIDA

		Infrastruktura i umiejętności IT	
Polityka IT		Silne	Słabe
	Doprecyzowane prawo	Wzmocnienie możliwości tworzenia w zakresie OS jako państwo, implementacja e-administracji, stworzenie rzecznictwa i zaplecza edukacyjnego.	Stworzenie rzecznictwa oraz zaplecza edukacyjnego oraz wzmocnienie możliwości tworzenia w zakresie OS jako państwo.
	Brak odpowiednich regulacji	Opracowanie polityki IT, utworzenie rzecznictwa oraz zaplecza edukacyjnego, wprowadzenie e-administracji oraz praca nad umocnieniem swojej pozycji na rynku międzynarodowym.	Opracowanie polityki IT, stworzenie rzecznictwa oraz zaplecza edukacyjnego. Wzmocnienie możliwości tworzenia w zakresie OS jako państwo.

Źródło: opracowanie własne za: S. Weerawarana, J. Weeratunga, dz. cyt., s. 46.

[44] Tamże, s. 41–43.

[45] Tamże, s. 43–44.

Na podstawie powyższej tabeli można zauważyć, jakie zależności zachodzą między tymi płaszczyznami. Dzięki temu możliwa jest diagnoza problemu, a na późniejszym etapie wypracowanie adekwatnych celów. Nie tylko dotyczy to kwestii tego, nad czym należy pracować, ale również podmiotów, które powinny być zaangażowane, kto powinien wspierać działania finansowo, merytorycznie itp. Tym samym otwierane są nowe możliwości rozwoju i inwestycji, które mogą pozytywnie wpłynąć na rozwój kraju na różnych poziomach.

Case-studies

W tym miejscu warto przytoczyć przykłady państw, dla których wprowadzenie zmian, związanych z wykorzystaniem rozwiązań OS, zakończyło się sukcesem.

Pierwszym z nich jest przypadek Sri Lanki, która posiadała odpowiednie zaplecze legislacyjne, ale także rozbudowaną infrastrukturę IT. Dzięki temu jej działania, bazując na wspomnianym powyżej grafie, powinny były skupić się na rozwoju e-administracji, wzmacnianiu pozycji państwowej na skalę międzynarodową oraz rozwój zaplecza edukacyjnego. Tym samym w kraju zostały wprowadzone pewne zmiany:

- stworzono agencję rządową odpowiedzialną za wprowadzanie rozwiązań OS;
- wdrożono działania mające na celu zupełne rozpowszechnienie Internetu;
- powołano grupy odpowiedzialne za korzystanie z Linux'a oraz innych OSS;
- prowadzono działania mające na celu rozbudowanie programów edukacyjnych oraz szkoleniowych na temat wykorzystania OSS;
- zainwestowano środki w rozwój centrów badawczych oraz rozwiązań OS, w celu wzmocnienia pozycji państwa na rynku;
- zaimplementowano rozwiązania w zakresie e-administracji, które poprawiły jakość skuteczności i zależności reagowania rządu[46].

Dla porównania można przybliżyć tutaj przypadek Iranu, który w 2019 roku był jednym z krajów, gdzie nastąpił największy wzrost liczby projektów OSS (o 44% w porównaniu z rokiem 2019)[47]. Jednakże na podstawie przeprowadzonych badań zauważono, że jego uwarunkowania nie są aż tak sprzyjające, jak w przypadku Sri Lanki. Pomimo dużego zaangażowania społecznego i umiejętności obywateli, Iran wciąż mierzy się z konsekwencjami braków na płaszczyźnie społeczno-ekonomicznej. Między innymi zostały tu wyróżnione problemy takie jak:

- brak doświadczenia w korzystaniu i budowaniu projektów OSS;
- brak wsparcia (na poziomie krajowym i międzynarodowym);
- problemy komunikacyjne między programistami a użytkownikami;
- niedostateczne wsparcie finansowe[48].

46 Tamże, s. 47–48.

47 S. Liu, *Leading countries/regions in terms of open source contributor growth rates in 2020*, źródło: https://www.statista.com/statistics/1068234/open-source-contributors-fast-growing-countries-regions/ [16.02.2021 r.].

48 A. Amrollahi, M. Khansari, A. Manian, *Success of Open Source in Developing Countries: The Case of Iran*, „International Journal of Open Source Software and Processes" 2014, no. 5(1), s. 50–65.

Odpowiednio przeprowadzona analiza działań państwa umożliwiła zwrócenie uwagi na to, jakie są mocne strony, ale też słabości wewnętrznych systemów krajowych. Dzięki temu dostosowywanie rozwiązań do potrzeb będzie skutkowało skuteczniejszą oraz szybszą implementacją. Poza tym należy zwrócić uwagę, że aktualnie wiele państw próbuje samodzielnie znaleźć sposób, w jaki mogą skorzystać z rozwiązań OSS w swoich systemach. Pozostaje jednak wciąż pytanie, czy ujednolicone odgórnie standardy lub systemy działań byłyby wsparciem w usprawnieniu całego procesu implementacji *open source software*.

Tytuł w języku angielskim:
USE OF OPEN-SOURCE SOFTWARE IN THE ACTIVITIES OF INTERNATIONALORGANIZATIONS ON THE EXAMPLE OF THE UNITED NATIONS AND THE EUROPEAN UNION

Bibliografia

Publikacje zwarte

Feller J., Fitzgerald B., *Understanding Open Source Software Development*, Addison-Wesley, London 2002.

Randhawa S., *Open Source Software and Libraries*, Centre For Research in Rural and Industrial Development, January 2008.

Raymond E.S., *The Cathedral and The Bazaar Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly 1999, s. 61–62.

Weber S., *Open Source Software in Developing Economies*, working paper, University of California, Berkeley 2003.

Weber S., *The Political Economy of Open Source Software*, BRIE Working Paper 140 Economy Project Working Paper 15 June 2000, s. 3.

Weerawarana S., Weeratunga J., *Open Source in Developing Countries*, SIDA, 2004.

Artykuły

Amrollahi A., Khansari M., Manian A, *How Open Source Software Succeeds? A Review of Research on Success of Open Source Software*, IJICTR, Volume 6, Number 2, Spring 2014.

Amrollahi A., Khansari M., Manian A., *Success of Open Source in Developing Countries: The Case of Iran*, International Journal of Open Source Software and Processes 2014, 5(1), 50–65, January–March.

Anand A., Krishna A., Tiwari R., Sharma R., *Comparative Analysis between Proprietary Software VS. Open--Source Software VS. Free Software*, 5th IEEE International Conference on Parallel, India 2018.

Banerjee P., *On the security of Open Source Software*, International Journal of Advanced Research, November 2017, s. 1339.

Delone W., McLean E., *The DeLone and McLean Model of Information Systems Success: A Ten-Year Update*, Journal of Management Information Systems, April 2003, s. 24.

Hars A., Ou S., *Work for free? Motivations for participating in Open-source projects*, International Journal of Electronic Commerce, Spring 2002, Vol. 6, No. 3, Communities in the Digital Economy.

Liu S., *Leading countries/regions in terms of open source contributor growth rates in 2020*, https://www.statista.com/statistics/1068234/open-source-contributors-fast-growing-countries-regions/ [luty 2021 r.].

Morgan L., Finnegan P., *Benefits and Drawbacks of Open Source Software: An Exploratory Study of Secondary Software Firms*, Limerick, Ireland 2007, s. 308.

Peterson C., *How I coined the term 'open source'*, https://opensource.com/article/18/2/coining-term-open-source-software [styczeń 2021 r.].

Źródła internetowe

Berkeley Internet Name Domain, źródło: https://www.bind9.net

Hamerly J., Paquin T., Walton S., źródło: https://www.oreilly.com/openbook/opensources/book/netrev.html.

History of the OSI, źródło: https://opensource.org/history.

Open Source Initiative, *The Open Source Definition*, źródło: https://opensource.org/osd.

Platforma GNU Operating System, *The GNU Operating System*, źródło: https://www.gnu.org/gnu/gnu.html

Platforma GNU Operating System, *What is Copyleft?*, źródło: https://www.gnu.org/licenses/copyleft.html.

Platforma GNU Operating System, *What is free software?*, źródło: https://www.gnu.org/philosophy/free-sw.html.

Platforma Open Source Initiative, *Frequently Answered Questions*, źródło: https://opensource.org/faq#osd.

Platforma Open Source Initiative, *Licenses by Name*, źródło: https://opensource.org/licenses/alphabetical.

Platforma Open Source Initiative, *The License Review Process*, źródło: https://opensource.org/approval.

Synopsys, *Open Source Security and Risk Analysis 2020*, https://www.synopsys.com/software-integrity/resources/analyst-reports/open-source-security-risk-analysis.html.