

Użycie oprogramowania Open Source – co poza „gift economy”?

Dorota Celińska*

Abstrakt

W artykule podjęto próbę określenia, jakimi powodami kierują się użytkownicy podczas wyboru oprogramowania Open Source. Wykorzystano analizę korespondencji, aby zbadać zależność pomiędzy używaniem programów z wybranych grup oprogramowania a deklarowanymi powodami ich wykorzystania zgodnymi z wymienianymi w literaturze. Dla porównania wyników z dostępną literaturą zastosowano również podział powodów zgodny z teorią Technology Acceptance Model (TAM). Otrzymane wyniki potwierdziły, że użytkownicy podczas wyboru oprogramowania nie kierują się jedynie niskim kosztem programów objętych licencją Open Source, a deklarowane powody korzystania mogą różnić się w zależności od grupy oprogramowania.

Słowa kluczowe: oprogramowanie, użytkownik, analiza korespondencji, powody użycia, Open Source

JEL Code: L17, L86, C38, D12.

* Wydział Nauk Ekonomicznych Uniwersytetu Warszawskiego.

Wstęp

Oprogramowanie Open Source stanowi klasę oprogramowania o dostępnym publicznie kodzie źródłowym aplikacji i możliwości jego modyfikowania przez użytkownika. Choć oprogramowanie tej klasy jest już przez długi czas dostępne na rynku oprogramowania, opis ekonomiczny tego zjawiska nadal cechują pewne braki.

Analizy Open Source koncentrują się na wytwarzaniu tego rodzaju oprogramowania. Autorzy skupiają uwagę na czynnikach wpływających na objęcie projektu licencją Open Source [Lerner i Tirole, 2005a; Koski, 2005]. Drugim popularnym tematem poruszonym w literaturze ekonomicznej są czynniki motywujące programistów do brania udziału w tego typu projektach [Schiff, 2002; Hann, 2004; Hars, 2002; Crowston et al., 2012]. Zainteresowanie badaczy opisanymi obszarami wynika z faktu, że oprogramowanie Open Source może być rozpatrywane przez pryzmat dóbr publicznych ze względu na szczególny sposób definiowania praw własności do kodu programu. Kod objęty licencją Open Source musi być dostępny publicznie oraz nie można wykluczyć nikogo z możliwości modyfikacji i korzystania z niego.

Pomimo dużej liczby artykułów podejmujących tematykę oprogramowania Open Source (np. Lerner i Tirole [2005b], Bonaccorsi i Rossi [2006], Gwembu i Wang [2010]) przyczyny wykorzystywania Open Source przez użytkowników końcowych praktycznie nie były do tej pory tematem analiz ekonomicznych. Niski poziom uwagi poświęcony użytkownikom Open Source wynika z potraktowania oprogramowania objętego tą licencją jako dobra publicznego lub popularnego, chociaż błędnego poglądu o ograniczeniu programów objętych tymi licencjami do bezpłatnych aplikacji [Gallego et al., 2008]. Oprogramowanie Open Source nie musi być oferowane za darmo – może być objęte tzw. podwójnym licencjonowaniem. Ten model licencjonowania zakłada jednoczesne rozpowszechnianie oprogramowania na dwóch różnych zestawach warunków. Jeden z produktów jest w tym modelu rozszerzony o płatne dodatki, których kod źródłowy nie jest udostępniany lub które mają za zadanie ułatwić integrację z prawnie zastrzeżonymi komponentami. Płatne mogą być również usługi związane z pielęgnacją i użytkowaniem udostępnionego kodu. Niski koszt oprogramowania Open Source nie musi być jednak jedynym czynnikiem motywującym użytkowników do korzystania z tego oprogramowania. Programiści są z dużym prawdopodobieństwem jednocześnie użytkownikami swoich produktów, chociaż nie każdy użytkownik posiada odpowiednią wiedzę, aby rozwijać projekt [Gacek i Arief, 2004]. W przypadku aktywnych użytkowników (współtworzących kod lub zgłaszających błędy w aplikacji) czynniki warunkujące decyzję o użyciu oprogramowania Open Source mogą być zbieżne z wymienionymi w literaturze czynnikami motywującymi programistów (np. zdobywane doświadczenie czy przekonanie o wyższej jakości oferowanych produktów [Hann, 2004; Crowston et al., 2012; Hars, 2002]). Ist-

nienie efektu sieciowego (zakładając nawet ograniczenie go do efektu sieciowego bezpośredniego, czyli wpływu innych osób) sprawia, że użytkownicy podejmują decyzje o użyciu programów Open Source również w oparciu o zachowanie pozostałych graczy na rynku oprogramowania. Decyzje użytkowników o wyborze oprogramowania wpływają na zachowania firm produkujących oprogramowanie, jednocześnie firmy produkujące oprogramowanie poprzez wykorzystanie efektu sieciowego pochodnego z produktów bądź usług komplementarnych dla oprogramowania wpływają na postawy użytkowników. Charakterystyka dobra publicznego i niski koszt może z jednej strony zachęcać do użycia oprogramowania Open Source, z drugiej w niektórych przypadkach wywoływać opór („gdyby było wysokiej jakości, nie byłoby dostępne bezpłatnie”). Wreszcie, społeczność (rozumiana jako zespół osób bezpośrednio zainteresowanych projektem w tym jego współtwórców) stanowi otoczenie instytucjonalne, oparte na potrzebach użytkowników, interakcjach między nimi, ale również przekonaniu, że dzielenie się informacją i wiedzą jest właściwym zachowaniem podczas tworzenia oprogramowania.

Celem tego artykułu jest wyodrębnienie powodów skłaniających użytkowników do wyboru oprogramowania Open Source i wykazanie, że niski koszt oprogramowania objętego tą licencją nie jest jedynym powodem skłaniającym użytkowników do jego wykorzystywania. Pod pojęciem „użytkowników” rozumie się tzw. użytkowników końcowych (ang. *end users*), czyli osoby fizyczne osobiście korzystające z legalnych kopii programów. Ze względu na niską liczbę artykułów poświęconych temu tematowi, niniejsza analiza ma charakter eksploracyjny. Zdecydowano się na użycie analizy korespondencji, dotychczas niewykorzystywanej techniki podczas badania użytkowników oprogramowania Open Source. Otrzymane wyniki potwierdziły występowanie innych niż niski koszt powodów korzystania z oprogramowania Open Source.

W części 1 artykułu przedstawiono krótki zarys dotychczasowych badań użytkowników oprogramowania Open Source. W części 2 opisano założenia wykorzystywanej techniki analizy. W 3 części opisany jest zbiór danych oraz zmienne wykorzystywane podczas analizy. W 4 części przedstawione zostały wyniki analizy a 5 sekcja stanowi podsumowanie.

1. Użytkownicy Open Source w dotychczasowych badaniach

Mimo że analizom użytkowników Open Source poświęcano mało uwagi podczas badań ekonomicznych, badacze byli zgodni, że użytkownicy mogą zostać pogrupowani ze względu na potrzeby i umiejętności informatyczne. Według Gacek [2004] wyróżnia się użytkowników pasywnych (ang. *passive*), niewnoszących poprawek do projektu oraz aktywnych (ang. *active*): programistów (ang. *developers*) i nie-programistów (ang. *nondevelopers*). Nie-programiści zgłaszają błędy (ang. *bugs*) i sugerują nowe funkcjonalności, ale w odróżnieniu od programistów nieedy-

tują bezpośrednio kodu. Programiści mogą dzielić się na wspomagających (ang. *code developers*) oraz głównych (ang. *core developers*). Decyzje o kierunku dalszego rozwoju projektu najczęściej należą do głównych programistów. Ten podział jednak nie będzie użyty w tym artykule z powodu trudności z dotarciem do reprezentatywnej liczby osób z każdej grupy ze względu na zbyt małą grupę głównych programistów projektów Open Source w Polsce.

Kolejny podział użytkowników programów Open Source, nie różnicujący ich ze względu na pełnione funkcje w społeczności, został zaproponowany przez Sena [2006]. W swoim badaniu popularności projektów Open Source dzieli on docelowych użytkowników aplikacji na 6 grup: użytkownicy zaawansowani (ang. *advanced users*), programiści (ang. *developers*), użytkownicy indywidualni (ang. *desktop users*), inżynierowie (ang. *quality engineers*), administratorzy (ang. *system administrators*) oraz użytkownicy niepasujący do żadnej z wymienionych grup (grupa „inne” – ang. *other*). Każda z grup posiada indywidualny poziom zaawansowania i oczekiwania wobec produktu.

Gagné [2005] wyróżnia mniej grup użytkowników oprogramowania Open Source niż Sen. Pierwszą grupę tworzą zaawansowani użytkownicy: inżynierowie i technicy, którzy prawdopodobnie znają GNU/Linuksa. Mają unikalne potrzeby i umiejętności oraz nie obawiają się zmiany używanej dystrybucji systemu operacyjnego czy środowiska pracy na inne. Druga grupa to mniej wymagający użytkownicy, potrzebujący narzędzi biurowych (edytora tekstu, arkusza kalkulacyjnego, przeglądarki internetowej...). To liczna grupa niekoniecznie technicznie zaawansowana. Trzecią grupę stanowią osoby, które wymagają programów objętych licencjami zamkniętego oprogramowania, dla których ciężko znaleźć jest odpowiedniki Open Source. Ta grupa użytkowników musi mieć dostęp do systemu operacyjnego MS Windows lub Mac OS X.

Jedno z podejść do opisu korzystania z oprogramowania Open Source przez użytkowników zostało opisane w artykule Gwembu i Wang [2010]. Badacze zastosowali teorię *Technology Acceptance Model* (TAM). Teoria TAM, z modyfikacjami wprowadzonymi przez autorów artykułu, opiera się na następujących założeniach:

- zachowanie użytkowników (decyzje o użyciu programów) wynika z wcześniejszej intencji do jego użycia (ang. *behavioral intention*);
- w podstawowej wersji teorii TAM (zaproponowanej przez Davisa w 1989 r.) intencje użytkowników co do użycia programu wynikają z dwóch przekonań użytkowników: postrzeganej przydatności (ang. *perceived usefulness*, czyli subiektywnej oceny użytkowników, na ile dany program będzie im potrzebny podczas pracy z komputerem) oraz postrzeganej łatwości użycia (ang. *perceived ease of use*, czyli subiektywnej oceny użytkowników, na ile łatwo będzie się nauczyć korzystania z danego programu);
- zmodyfikowana wersja teorii TAM dopuszcza więcej determinant intencji

- użytkowników co do użycia programu: postrzeganą zgodność z dotychczasowym doświadczeniem (ang. *perceived compatibility with prior experience*, czyli subiektywną ocenę użytkowników, na ile obsługa danego programu przypomina te, z którymi mieli już do czynienia), postrzeganą zgodność z filozofią oprogramowania (ang. *perceived compatibility with software philosophy*, czyli subiektywną ocenę użytkowników, na ile użycie danego programu będzie zgodne z wyznawanymi przez użytkowników zasadami dotyczącymi użycia i tworzenia oprogramowania), postrzegane ryzyko związane z użyciem danego programu (ang. *perceived risks*), czyli subiektywną ocenę ryzyka związanego z użyciem danego programu: np. obawy użytkowników o jakość i częstotliwość jego aktualizacji);
- w zmodyfikowanej wersji teorii TAM autorzy uzależnili intencje użycia nowego programu od dotychczasowego zadowolenia (ang. *satisfaction*) z innych programów Open Source używanych przez użytkowników.

Gwembu i Wang przebadali powody używania oprogramowania Open Source w czterech segmentach użytkowników. Pierwszy segment stanowił potencjalny rynek głównego nurtu (ang. *potential mainstream market*) – użytkownicy nienależący do społeczności Open Source i niebędący programistami, zwykli użytkownicy ze słabą znajomością technologii informatycznej wymagający programów do rutynowych czynności. Drugi segment obejmował potencjalny rynek niszowy (ang. *potential niche market*), czyli użytkowników nienależących do społeczności Open Source, ale będących programistami, którzy mają wysoką wiedzę z zakresu technologii informatycznych. Do trzeciego segmentu należały osoby ze spenetrowanego rynku głównego nurtu (ang. *penetrated mainstream market*), będący użytkownikami należącymi do społeczności Open Source, ale niebędącymi programistami, zwykli użytkownicy ze słabą znajomością technologii informatycznej, ale z doświadczeniem w używaniu oprogramowania Open Source. Ostatni segment stanowił spenetrowany rynek niszowy (ang. *penetrated niche market*), obejmujący użytkowników należących do społeczności Open Source i będących programistami, czyli osoby z zaawansowaną wiedzą z zakresu technologii informatycznej, które mają również doświadczenie w używaniu oprogramowania Open Source. Dane do badania zostały uzyskane za pomocą internetowego badania ankietowego poprzedzonego pilotażem. Pytania w kwestionariuszu oparto na 5-stopniowej skali Likerta. Badanie zostało skierowane do dwóch grup użytkowników: uczestników konferencji Ohio LinuxFest (125 osób) oraz wylosowanych studentów Ohio State University (200 osób). Otrzymano 168 całkowicie wypełnionych kwestionariuszy, tworzących próbę zawierającą 25,6% kobiet i 73,8% mężczyzn o średniej wieku 27,62 lat. 38% próby stanowili użytkownicy potencjalnego rynku głównego nurtu; 19% próby należało do potencjalnego rynku niszowego; 24% próby stanowiło spenetrowany rynek głównego nurtu, a 19% próby należało do spenetrowa-

negu rynku niszowego. Za pomocą testu Manna-Whitneya wskazano statystycznie istotne różnice w postrzeganiu potencjalnej przydatności oraz potencjalnego ryzyka wynikającego z użycia oprogramowania Open Source pomiędzy grupami programistów i nie-programistów. Wykazano również brak statystycznie istotnych różnic w postrzeganiu potencjalnej przydatności i zgodności z dotychczasowym doświadczeniem w grupach członków społeczności a także osób nienależących do niej. Użytkownicy należący do potencjalnego rynku głównego nurtu wykazywali statystycznie istotnie niższe wartości charakterystyk opisujących postrzeganą łatwość użycia oprogramowania, satysfakcji z niego oraz wyższe wartości charakterystyk opisujących potencjalne ryzyko związane z używaniem oprogramowania Open Source niż użytkownicy spenetrowanego rynku niszowego. Użytkownicy potencjalnego rynku głównego nurtu postrzegali oprogramowanie Open Source jako mniej zgodne z ich filozofią oprogramowania oraz dające niższą satysfakcję z użycia niż użytkownicy należący do spenetrowanego rynku głównego nurtu. Użytkowników spenetrowanego niszowego rynku cechowały statystycznie istotnie wyższe wartości charakterystyk opisujących łatwość użycia oprogramowania i satysfakcję z niego w porównaniu do użytkowników potencjalnego niszowego rynku. Niestety, ze względu na niską reprezentatywność użytej próby otrzymane przez autorów wnioski nie mogą być uogólniane na wszystkich użytkowników oprogramowania Open Source na świecie.

W artykule Gallego, Luny i Bueno [2008] przeprowadzono badanie czynników wpływających na podjęcie decyzji o użyciu oprogramowania Open Source przez użytkowników w oparciu o podstawową teorię TAM. Badanie zostało oparte na próbie pochodzącej z badania ankietowego wykorzystując kwestionariusz internetowy rozesłany do 1736 użytkowników GNU/Linuksa zarejestrowanych na stronie Linux Counter, poprzedzony pilotażem. Otrzymano 347 całkowicie wypełnionych kwestionariuszy (stopa zwrotu, ang. *response rate* na poziomie 20%). W kwestionariuszu uwzględniono pytania oparte na 7-stopniowej skali Likerta odpowiadające następującym elementom teorii TAM:

- postrzeganej przydatności, postrzeganej łatwości użycia, intencji użycia oraz deklarowanemu zachowaniu użytkowników (ang. *usage behaviour*) oprogramowania Open Source;
- jakości i zdolnościom (ang. *capability*) oraz elastyczności oprogramowania Open Source (szybkość działania programu, możliwość dostosowania go do potrzeb użytkowników);
- wpływowi środowiska i znajomych użytkowników (ang. *social influence*) na decyzje o użyciu oprogramowania Open Source.

Gallego, Luna i Bueno do analizy determinant podjęcia decyzji o użyciu oprogramowania Open Source przez użytkowników wykorzystali model wielorównaniowy. Na jego podstawie określono, iż postrzegana łatwość użycia

oprogramowania Open Source miała statystycznie istotny i pozytywny wpływ na postrzeganą przydatność i intencje użycia programów objętych tą licencją. Postrzegana przydatność oprogramowania Open Source pozytywnie i statystycznie istotnie wpływała na intencje użytkowników do użycia oprogramowania Open Source. Intencje użytkowników do użycia oprogramowania Open Source wykazywały pozytywny i statystycznie istotny wpływ na deklarowane zachowanie użytkowników. Postrzegana przydatność oprogramowania Open Source pozytywnie i statystycznie istotnie zależała od postrzeganej przez użytkowników jakości oraz elastyczności programów objętych tą licencją. Wpływ środowiska i znajomych użytkowników nie miał statystycznie istotnego wpływu na postrzeganą przydatność i łatwość użycia oprogramowania Open Source.

Ze względu na niewielką liczbę artykułów zawierających wyniki badań empirycznych poświęconych problematyce podjęcia decyzji o użyciu oprogramowania Open Source przez użytkowników, w niniejszym artykule odwołano się do teoretyków i założycieli Open Source Initiative. Perens [2013], autor Umowy Społecznej Debiana, jako pierwszy podkreślił wyższą jakość produktów Open Source w stosunku do programów objętych licencjami zamkniętego oprogramowania. Ze względu na fakt, że rozwojem projektów Open Source zajmują się użytkownicy programów, błędy występujące w aplikacjach są szybko znajdowane i naprawiane. W przeciwieństwie do programów objętych licencjami zamkniętego oprogramowania, oprogramowanie Open Source jest obciążone niższym ryzykiem zniknięcia z rynku oprogramowania (poza młodym stadium, łatwo znajdują osoby chętne, by je poprowadzić, a nawet, gdy takich brak, kody źródłowe są ciągle dostępne i gotowe do rozwoju). Dołączanie do aplikacji funkcjonalności wiąże się z niższym kosztem w porównaniu z dołączeniem funkcjonalności do programów objętych licencjami zamkniętego oprogramowania z powodu łatwiejszego dotarcia do programistów (są równocześnie użytkownikami współtworzonego produktu).

Raymond [2004], jeden z głównych założycieli Open Source Initiative jako jedną z cech przemawiających za użyciem oprogramowania Open Source wymienia błyskawiczny rozwój projektów dzięki istnieniu społeczności. Aplikacje objęte licencją Open Source cechuje elastyczność i rozszerzalność. Komponenty mogą być łączone na wiele sposobów, programy wykonujące drobne zadania mogą zostać wykorzystane w większych projektach. Kolejną cechą skłaniającą ku wyborowi oprogramowania Open Source jest prostota interfejsów programów i niezależność od konkretnego modelu sprzętu. Programy są względnie nieduże, przejrzyste, zgodne z „regułą najmniejszego zaskoczenia”. Produkty charakteryzuje również stabilność i działanie w niespodziewanych warunkach. Użycie oprogramowania Open Source umożliwia również zdobycie doświadczenia przez użytkowników, zarówno podczas czytania kodów źródłowych, jak i zabawy oprogramowaniem. Wymienione cechy wynikają w dużej mierze z treści licencji Open Source i dostępu użytkowników do kodów źródłowych aplikacji. Zjawisko Open Source okre-

ślane jest mianem „ekonomii prezentu” (ang. *gift economy*) [Raymond, 2012]. Ekonomia prezentu polega na wykształceniu się struktur i zachowań jednostek w warunkach nadmiaru a nie braku, jak w klasycznej ekonomii wymiany. Powstaje w populacjach mających wystarczającą ilość dóbr wymaganych przez te populacje do przeżycia. W przypadku społeczności Open Source dobrami wymaganymi do przeżycia są przestrzeń dyskowa, przepustowość sieci lub moc obliczeniowa komputerów. W obecnych czasach nie stanowią one ograniczenia. Z tego powodu użytkownicy oprogramowania Open Source mogą się między sobą dzielić a nie wymieniać kodami źródłowymi aplikacji. Użytkownicy niebędący programistami otrzymują od twórców „w prezencie”, nieodpłatnie, sprawny program. Programiści Open Source rywalizują o reputację w społeczności. Chociaż programiści Open Source powinni kierować się zasadą, że tworzenie upublicznionego kodu i dzielenie się wiedzą jest właściwym postępowaniem podczas tworzenia i korzystania z oprogramowania, zachowanie twórców kodu i praktyczne powody rozwijania i korzystania z oprogramowania Open Source różnią się od powodów wynikających z treści licencji Open Source.

Podobnie jak Raymond, inny z założycieli *Open Source Initiative*, Allman zauważa, iż projekty Open Source szybko rozwijają się dzięki łatwemu dostępowi do grupy testerów. Częstotliwość aktualizacji nie jest jednak główną determinantą użycia oprogramowania Open Source, ponieważ nie zawsze testerzy należą do osób kompetentnych. Według Allmana dla zwykłych użytkowników, czyli takich, których cechuje niski poziom umiejętności technicznych, przy podejmowaniu decyzji o wyborze oprogramowania nie ma znaczenia objęcie aplikacji licencją Open Source i wynikający z niej dostęp do kodów źródłowych. Dla użytkowników ważny jest niski koszt oprogramowania Open Source [Allman, 2012].

Wśród osób identyfikowanych z założycielami Open Source są również twórcy języków programowania: van Rossum (Python) oraz Wall (Perl). Według van Rossuma zaletami Open Source mogącymi wpłynąć na wybór tego oprogramowania przez użytkowników są jakość oprogramowania wynikająca z dużej liczby osób czytających kody źródłowe i wyłapujących w nich błędy oraz możliwość modyfikacji aplikacji przez naniesienie poprawek do ich udostępnionych kodów źródłowych [Rossum, 2013]. Wall jako powód istotnie wpływający na decyzję o wyborze Perla (objętego licencją Open Source) przez użytkowników podkreśla zasadę, którą kierował się podczas tworzenia tego języka – korzystając z Perla, użytkownicy nie są ograniczeni do pojedynczych, konkretnych sposobów rozwiązywania danego problemu, mogą np. wybrać spośród różnych zapisów tej samej funkcjonalności. Dodatkowo, chociaż Perl 5 jest głównie dziełem Walla, Perl 6 stanowi język programowania z dużym udziałem społeczności w procesie tworzenia, dzięki czemu jego składnia uwzględnia potrzeby większej liczby osób.

Gagné [2005] wspomina o niskim koszcie rozwiązań Open Source, nie sprowadzając go do „darmowego obiadu”. Zawsze istnieją pewne koszty: np. łącze

internetowe potrzebne do pobrania oprogramowania, czy czas poświęcony przez użytkowników na instalację i naukę obsługi nowego programu. Podkreśla prawny aspekt użycia aplikacji Open Source. Podobnie jak dla Raymonda i van Rossuma mocną cechą Open Source stanowi dla niego stabilność programów Open Source, czyli to, że niespodziewane błędy programów są rzadkością. Dodatkowymi zaletami oprogramowania Open Source są elastyczność jego konfiguracji oraz obecność społeczności. Objęcie programu licencją Open Source wiąże się według Gagné z brakiem tajemnic, zarówno dla użytkowników (wiedzą, co dzieje się i znajduje w produkcie), jak i ze strony producentów oprogramowania (nie tuszują i nie ukrywają problemów, a poprawiają je).

Sokół [2006] w książce poświęconej jednej z najstarszych dystrybucji GNU/Linux (Slackware) opisuje darmowy i jednocześnie legalny charakter tego systemu operacyjnego. Slackware GNU/Linux jest często aktualizowany i odporny na wirusy. Cechuje go stabilność, wydajność oraz wysoka konfigurowalność, polegająca na możliwości wyboru wielu środowisk graficznych. Ze względu na dostęp użytkowników do kodu źródłowego, nie występuje efekt „wiem, jak coś zrobić, lecz nie wiem, czemu tak się dzieje”. Według głównych *developerów* systemu operacyjnego Archlinux, czyli osób mających bezpośredni wpływ na decyzje związane z kierunkiem rozwoju systemu operacyjnego, tworzona przez nich dystrybucja GNU/Linux jest systemem operacyjnym wysoce konfigurowalnym, przy jednoczesnej prostocie wprowadzania modyfikacji z punktu widzenia użytkowników, zgodnie z zasadą *Keep it Simple* [Archlinux, 2013]. System operacyjny aktualizowany jest na bieżąco – każda aktualizacja pojedynczego programu jest natychmiast dostępna do pobrania przez użytkowników.

Powody używania oprogramowania Open Source w Polsce stanowiły przedmiot badań, których wyniki przedstawiono w raporcie Pentor Research International [2010]. Badanie dotyczyło rządowej administracji publicznej w Polsce, dane zbierane były metodą CAWI (*Computer Assisted Web Interview*). Badanie ankietowe przeprowadzone w lutym 2010 roku było skierowane do pracowników działów informatycznych administracji rządowej szczebla centralnego i wojewódzkiego. Z 500 wysłanych zaproszeń do badania ankietowego, otrzymano 101 całkowicie wypełnionych kwestionariuszy ankiet (*response rate* na poziomie 20%). Najczęstszymi powodami użycia oprogramowania Open Source były według ankietowanych wysokie koszty oprogramowania objętego licencjami zamkniętego oprogramowania i jego zbyt wysokie ceny (69% ankietowanych), przekonanie o jakości oprogramowania Open Source nie odbiegającej od jakości oprogramowania objętego licencjami zamkniętego oprogramowania (40% ankietowanych) oraz przekonanie o tym, że oprogramowanie Open Source nie prowadzi do wykluczenia cyfrowego jego użytkowników (29% ankietowanych). Najrzadszymi powodami korzystania z oprogramowania Open Source były: dostęp do kodu źródłowego (16% ankietowanych), stabilność programów objętych licencją Open Source

(14% ankietowanych) oraz wymóg zastosowania oprogramowania Open Source, np. ze względu na zalecenie urzędu wyższej rangi (10% ankietowanych).

2. Założenia analizy korespondencji

Analiza korespondencji należy do technik analizy wielowymiarowej danych jakościowych. Technika analizy polega na przeprowadzeniu operacji na tabeli wielodzielczej – tabeli, która przedstawia rozkład obserwacji ze względu na kilka zmiennych mających różne kategorie jednocześnie. Analiza korespondencji stanowi technikę dostarczającą informacji na temat struktury powiązań pomiędzy wierszami i kolumnami tabeli wielodzielczej. Graficzna prezentacja wyników tej techniki analizy umożliwia intuicyjne wnioskowanie odnośnie powiązań zachodzących pomiędzy kategoriami badanych zmiennych.

Analizę korespondencji rozpoczyna się od obliczenia macierzy korespondencji na podstawie tablicy wielodzielczej. Macierz korespondencji powstaje poprzez podzielenie liczebności w poszczególnych komórkach tabeli wielodzielczej przez liczebność całkowitą badanej próby. Kolejny krok stanowi wyznaczenie macierzy profili wierszowych i kolumnowych. Macierz profili wierszowych otrzymuje się, dzieląc poszczególne elementy wierszy macierzy korespondencji przez sumę wszystkich elementów tego wiersza. Macierz profili kolumnowych wyznaczana jest poprzez dzielenie poszczególnych elementów kolumn macierzy korespondencji przez sumę wszystkich elementów tej kolumny. Masą wiersza nazywa się sumę elementów danego wiersza macierzy korespondencji – kolumna o elementach będącymi masami wierszy tworzy przeciętny profil kolumnowy. Masa kolumny definiowana jest jako suma elementów danej kolumny macierzy korespondencji – wiersz o elementach będących masami kolumn tworzy przeciętny profil wierszowy. Analiza otrzymanych profili (zarówno wierszowych, jak i kolumnowych) odbywa się poprzez wyznaczenie odległości między nimi za pomocą metryki chi-kwadrat.

W przypadku analizy korespondencji terminem odpowiadającym pojęciu wariancji jest termin „bezwładności”. Bezwładność stanowi miarę rozproszenia profili wokół odpowiednich przeciętnych profili. Całkowita bezwładność kolumn wskazuje, jak mocno poszczególne profile kolumnowe są różne od przeciętnego profilu kolumnowego. Ponieważ bezwładność dla kolumn jest równa bezwładności dla wierszy, podaje się tylko jedną wartość nazywaną inercją całkowitą. Mniejsza bezwładność wiąże się z mniejszą szansą wystąpienia istotnego statystycznie związku pomiędzy kategoriami zmiennych w tabeli wielodzielczej. Dodatkowo, na podstawie bezwładności można określić liczbę wymiarów niezbędnych do odtworzenia maksimum informacji z próby. W przypadku dwóch wymiarów za zadowalające uznaje się odtworzenie co najmniej 75% inercji całkowitej.

Podczas interpretacji wyników analizy korespondencji należy określić poło-

zenie punktów obrazujących kategorie z wierszy i kolumn tabeli wielodzielczej. Określenie położenia punktów opiera się na zbadaniu ich rozmieszczenia względem centrum rzutowania, rozmieszczenia punktów odpowiadających kategoriom tej samej zmiennej względem siebie oraz rozmieszczenia punktów odpowiadających kategoriom różnych zmiennych względem siebie [Stanimir, 2005]. Kategorie zmiennych położone na wykresie w niedalekiej odległości od siebie wskazują kombinacje pojawiające się częściej niż jest to spodziewane przy założeniu niezależności wierszy i kolumn.

3. Zbiór danych oraz wykorzystane zmienne

Dane wykorzystywane w tym badaniu zostały zebrane za pomocą ankiety internetowej. Populację statystyczną badania stanowili użytkownicy Internetu z Polski w wieku 13–54 lat. Ograniczenia wiekowe nałożono, aby uniknąć potencjalnych problemów wynikających ze zbyt niskiej liczebności osób starszych wśród użytkowników Internetu. Dane zbierano od sierpnia 2012 roku do lutego 2013 roku. Badanie zostało poprzedzone pilotażem w lipcu 2012 roku.

Otrzymano 1713 całkowicie wypełnionych kwestionariuszy spośród 2412 nadesłanych kwestionariuszy. Badanie podzielone było na dwie grupy: grupę osób badanych za pomocą linków wysyłanych w emailach oraz grupę użytkowników forów i portali internetowych samodzielnie odwiedzających stronę internetową zawierającą kwestionariusz. Nie jest możliwe oszacowanie dokładnego *response rate* dla podgrupy osób odwiedzających strony internetowe, na których umieszczono link do kwestionariusza. W podpróbie osób pochodzących z list mailingowych *response rate* wyniósł 45%. Po usunięciu obserwacji spoza przedziału wiekowego 13–54 w próbie pozostało 1694 całkowicie wypełnionych kwestionariuszy.

Otrzymana próba nie była reprezentatywna, dlatego postanowiono zastosować wagi częstotliwościowe. Dzięki ich zastosowaniu otrzymano próbę reprezentatywną ze względu na wiek, płeć, wielkość miejsca zamieszkania oraz poziom wykształcenia respondenta.

Do badania powiązań pomiędzy korzystaniem z określonych grup oprogramowania a powodami korzystania deklarowanymi przez respondentów wybrano 9 grup oprogramowania:

- *biurowe* – 1, jeśli ankietowany deklaruje użycie programu z grupy: Libre/Open Office, Mozilla Firefox, Thunderbird, Chromium (Google Chrome), Claws Mail, 7zip, Pidgin, Android; 0, jeśli żadnego z wymienionych.
- *hobby* – 1, jeśli ankietowany deklaruje użycie z grupy: GIMP, Inkscape, Audacity, Amarak, VLC, Audacious, Rhythmbox, Blender, MPlayer; 0, jeśli żadnego z wymienionych.
- *systemy* – 1, jeśli ankietowany deklaruje użycie systemu operacyjnego z grupy: GNU/Linux, BSD, Open Solaris; 0, jeśli żadnego z wymienionych.

- *strony* – 1, jeśli ankietowany deklaruje użycie programu z grupy: Roundcube, SquirrelMail, Drupal, Wordpress, Joomla, Piwik/OWA, Mediawiki, PhpMyAdmin; 0, jeśli żadnego z wymienionych.
- *serwery* – 1, jeśli ankietowany deklaruje użycie programu z grupy: Apache, Nginx, lighttpd, Filezilla, PuTTY, MySQL, PostgreSQL, MongoDB, Virtualbox; 0, jeśli żadnego z wymienionych.
- *gry* – 1, jeśli ankietowany deklaruje użycie z grupy: Open Arena, Enemy Territory, 0AD, freeciv, Teeworlds, Tuxracer/SuperTux, Battle for Wesnoth; 0, jeśli żadnego z wymienionych.
- *programistyczne1 (dev1)* – 1, jeśli ankietowany deklaruje użycie programu z grupy: Eclipse, Netbeans, Geany, Emacs, gedit, ViM, GCC, CLISP/SBCL, GHC/Hugs; 0, jeśli żadnego z wymienionych.
- *programistyczne2 (dev2)* – 1, jeśli ankietowany deklaruje użycie programu z grupy: Git, bazaar, Mercurial, SVN, PHP, Perl, Django, Python, Ruby; 0, jeśli żadnego z wymienionych.
- *nauka* – 1, jeśli ankietowany deklaruje użycie programu z grupy: Maxima, R, Octave, arduino, gretl, gnuplot, Weka; 0, jeśli żadnego z wymienionych.

Kategorie zmiennej *powody* zostały wybrane w oparciu o dostępną literaturę. Początkowo podzielono zmienną na 13 kategorii, następnie poszczególne kategorie dopasowano do kategorii pochodzących ze zmodyfikowanej teorii *Technology Acceptance Model* (TAM) [Gwembu i Wang, 2010; Gallego et al., 2008]: postrzeganej przydatności (ang. *perceived usefulness*: PU); postrzeganej łatwości użycia (ang. *perceived ease of use*: PEU); zgodności z dotychczasowym doświadczeniem (ang. *compatibility with prior experience*: CPE); postrzeganej zgodności z filozofią oprogramowania (ang. *perceived compatibility with software philosophy*: SF) oraz zadowoleniem z użycia oprogramowania (ang. *satisfaction*: S).

Do postrzeganej przydatności (PU) zakwalifikowano kategorie: *popularny* (program jest popularny wśród znajomych i rodziny ankietowanego); *darmo* (program jest oferowany bezpłatnie) i *szybkość* (program szybko się rozwija). Jako nośniki postrzeganej zgodności z filozofią oprogramowania (SF) zaproponowano kategorie: *legalny* (program jest legalny, nie potrzeba dodatkowych osobnych umów i postanowień licencyjnych); *rozwój_wpływ* (użytkownik ma wpływ na rozwój projektu, np. poprzez zgłaszanie lub poprawianie błędów) oraz *uczenie się* (korzystając z programu użytkownik może się czegoś nauczyć, również poprzez czytanie i recenzowanie kodu źródłowego). Postrzegana łatwość użycia (PEU) objęła kategorie: *stabilność* (program działa dobrze w niespodziewanych warunkach, jest odporny na wirusy) i *prostota* (program jest prosty w obsłudze). Zgodność z dotychczasowym doświadczeniem (CPE) została zobrazowana w kategoriach *przyzwyczajenie* (ankietowany używa programu z przyzwyczajenia) oraz *wymagany* (program używany na uczelni, w szkole lub pracy). Zadowolenie (S) zostało

opisane kategoriami: *lepszy* (ankietowany uważa, że program jest lepszy od znanych mu odpowiedników); *grafika* (ankietowanemu podoba się wygląd programu – grafika, interfejs użytkownika) oraz *grywalność* (ankietowany lubi grać w grę).

Wartości tablicy wielozdzielczej zostały utworzone na podstawie następującego algorytmu. W przypadku zaznaczenia przez ankietowanego jakiegokolwiek odpowiedzi innej niż „nie używam żadnego” w pytaniach dotyczących użycia programów z 8 grup („proszę zaznaczyć programy, z których Pan (i) korzysta (lub zdarza się korzystać)”), spośród zaznaczonych przez ankietowanego programów w sposób losowy wybierany był „reprezentant” danej grupy oprogramowania. Następnie ankietowany pytany był o powody używania „reprezentanta” danej grupy oprogramowania. Przyjęto założenie, że zakreślone przez ankietowanego powody korzystania z „reprezentanta”, z którego deklaruje korzystanie, dobrze opisują powody używania grupy oprogramowania, zadawanie pytań o każdy program zaznaczony przez ankietowanego mogłoby doprowadzić do dużej liczby pytań i być męczące dla ankietowanego. Odpowiedzi ankietowanych (1, jeśli wskazali dany powód; 0 w przeciwnym przypadku) zostały zsumowane w ramach grupy oprogramowania, z podziałem na kategorie zmiennej *powody*. Podczas sumowania uwzględniono policzone wagi częstotliwościowe.

W tabeli 1 przedstawiono podstawowe charakterystyki opisowe dotyczące odpowiedzi ankietowanych na pytanie o powody używania „reprezentanta” grupy oprogramowania. Dla większości grup oprogramowania, mimo że do odpowiedzi na pytanie wystarczyło zaznaczyć jeden powód, ponad połowa ankietowanych zaznaczyła co najmniej 3 powody (wyjątki stanowią grupy *gry* z 2 powodami oraz *systemy* z 6 powodami). Poza grupą *systemy* rozkłady liczby odpowiedzi na pytania o powody używania „reprezentanta” grupy oprogramowania cechowała asymetria prawo-skośna (wartości średnich większe niż wartości median). Jest to zgodne z intuicją – większość osób udziela niewielkiej liczby odpowiedzi na pytania wielokrotnego wyboru.

Tabela 1: Charakterystyki opisowe liczby odpowiedzi ankietowanych na pytania dotyczące powodów używania „reprezentanta” grupy oprogramowania

Zmienna	Minimum	Maximum	Średnia	Mediana
biurowe	1	12	4,18	4
hobby	1	12	3,47	3
systemy	1	11	5,54	6
strony	1	11	3,29	3
serwery	1	10	3,39	3
gry	1	9	2,80	2
dev1	1	11	3,66	3
dev2	1	11	3,39	3
nauka	1	11	2,99	3

Źródło: Opracowanie własne.

W tabeli 2 została przedstawiona tabela wielodzielcza będąca podstawą przeprowadzanej analizy korespondencji dla całości badanej próby. Ankietowani udzielili (po zastosowaniu wag częstotliwościowych) 375521 odpowiedzi. Różnica w liczebności w porównaniu do badanej próby wynikała z możliwości udzielenia przez ankietowanych więcej niż jednej odpowiedzi na pytanie o powody użycia „reprezentanta” grupy oprogramowania. Najczęściej wymienianym powodem niezależnie od grupy oprogramowania była *darmowość* oprogramowania (tabela 2). Był to również najczęściej wymieniany powód wśród odpowiedzi w ramach grupy oprogramowania poza grupą *gry*, dla której popularniejszym powodem była *grywalność*. Dwa następne najpopularniejsze powody niezależnie od grupy oprogramowania to *legalny* oraz *prosty w obsłudze*, ta hierarchia nie była zachowana jednak w ramach wszystkich grup oprogramowania. Najmniej popularnymi powodami niezależnie od grupy oprogramowania były *przyzwyczajenie*, *program wymagany* oraz *możliwość wsparcia rozwoju programu*.

Tabela 2: Tablica wielodzielcza dla całości badanej próby

X	biurowe	hobby	systemy	strony	serwery	gry	dev1	dev2	nauka
popularny	7196	3965	1219	2580	2161	1268	1994	2457	301
darmo	14030	12922	9837	5158	7313	4398	5152	3858	1639
legalny	11481	10405	9800	4148	6004	3415	4846	3684	1526
uczenie	1504	3119	8449	2457	2281	0	2803	3119	1198
rozwój szyb	5153	2077	6018	1279	905	315	1395	1189	507
wymagany	1612	694	1154	1279	1426	0	1245	1294	780
rozwój wpływ	1960	1226	4374	931	494	289	827	685	399
stabilny	8143	6210	9814	2629	4239	961	3668	2266	807
lepszy	9064	7239	7320	2118	3611	1164	3073	1999	1047
grafika	6460	3219	5153	2545	4707	2501	2732	983	464
prostota	14382	9210	4627	4347	2390	0	4734	1893	1099
przyzwyczajenie	10781	5601	0	0	0	0	0	0	0
grywalność	0	0	0	0	0	4840	0	0	0

Źródło: Opracowanie własne.

W tabeli 3 przedstawiono tabelę wielodzielczą dla całości badanej próby, ale dla analizy powodów zgodnych z teorią TAM. Najczęściej wymienianymi przez ankietowanych powodami niezależnie od grupy oprogramowania były powody związane z postrzeganą przydatnością oprogramowania (ang. *perceived usefulness* PE). Najrzadziej ankietowani udzielali odpowiedzi dotyczących zadowolenia z używanego oprogramowania (ang. *satisfaction* S).

Tabela 3: Tabela wielozdzielcza dla całości badanej próby (powody zgodne z teorią TAM)

X	PU	SF	CPE	PEU	S
biurowe	26379	14945	12393	22525	15524
hobby	18964	14750	6295	15420	10458
systemy	17074	22623	1154	14441	12473
strony	9040	7575	1510	6976	4663
serwery	10379	8779	1426	6629	8312
gry	5981	3704	0	2125	7343
programistyczne1	8541	8476	1245	8402	5805
programistyczne2	7504	7488	1294	4159	1511
nauka	2447	3123	780	1906	2982

Zródło: Opracowanie własne.

4. Rezultaty i interpretacja

Przed przystąpieniem do analizy korespondencji sprawdzono, czy można użyć tej techniki analizy. Przeprowadzono test niezależności Chi-kwadrat. Poza niezależnością zmiennych *powody* oraz *grupa* sprawdzona została również siła związku między badanymi zmiennymi – użyto współczynnika zbieżności V-Cramera ze względu na badanie zmiennych nominalnych. Testy zostały przeprowadzone przy przyjęciu poziomu istotności 5%.

Na podstawie otrzymanych wyników testów można stwierdzić, że istnieją podstawy do odrzucenia hipotezy zerowej o niezależności badanych zmiennych (*p-value* testów Chi-kwadrat: $0,000 < 0,05$). Wartość współczynnika V-Cramera informuje, że między badanymi zmiennymi występuje słaby związek (wartości poniżej 0,30: odpowiednio 0,228 dla analizy dla 13 kategorii zmiennej *powody* oraz 0,224 dla analizy na kategoriach zmiennych zgodnych z teorią TAM). Ponieważ między zmiennymi zachodzi zależność można użyć analizy korespondencji.

W tabeli 4 przedstawiono wyniki analizy korespondencji dla 13 kategorii zmiennej *powody*. Inercja dla całości badanej próby wyniosła 0,4129. Dwa wymiary tłumaczą 88,01% całkowitej inercji, zatem analiza korespondencji dobrze tłumaczy zależności między badanymi zmiennymi. Na podstawie danych zgromadzonych w tabeli 4 można dostrzec, że największy wkład w całkowitą inercję mają grupy oprogramowania Open Source: *gry* (59,9% inercji), *biurowe* (12,4% inercji) oraz *systemy* (11,0% inercji). Spośród kategorii zmiennej *powody* największy wkład w inercję mają kategorie *grywalność* (58,1% inercji), *przyzwyczajenie* (15,2% inercji) oraz *uczenie się* (7,0% inercji). Największy wpływ na pierwszy wymiar ma grupa oprogramowania Open Source *gry* oraz kategoria zmiennej *powody*: *grywalność*. Na drugi wymiar największy wpływ mają grupy oprogramo-

wania Open Source: *biurowe* i *systemy* oraz kategorie zmiennej *powody: uczenie się* oraz *przyzwyczajenie*.

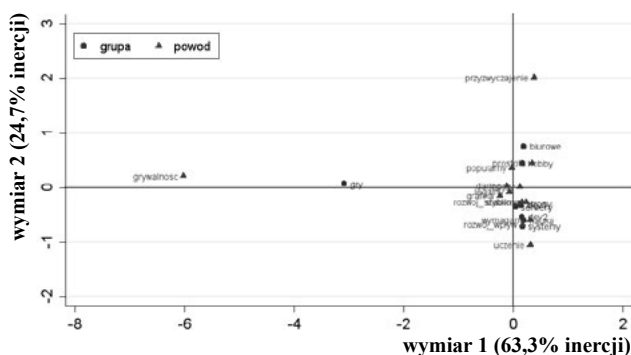
Tabela 4: Wyniki analizy korespondencji dla 13 kategorii zmiennej powody

Kategorie	Ogólnie		Wymiar 1	Wymiar 2
	Jakość wytłumaczenia	% inercji wyjaśnionej	Wkład	Wkład
grupa				
biurowe	0,957	0,124	0,020	0,430
hobby	0,766	0,043	0,010	0,106
systemy	0,708	0,110	0,011	0,289
strony	0,340	0,022	0,003	0,022
serwery	0,266	0,036	0,000	0,038
gry	1,000	0,599	0,946	0,001
dev1	0,567	0,017	0,004	0,029
dev2	0,494	0,033	0,003	0,056
nauka	0,538	0,016	0,002	0,030
powody				
popularny	0,261	0,024	0,000	0,025
darmo	0,342	0,009	0,005	0,000
legalny	0,436	0,004	0,002	0,003
uczenie	0,926	0,070	0,013	0,231
rozwój_szybkość	0,273	0,022	0,005	0,011
wymagany	0,336	0,030	0,005	0,028
rozwój_wpływ	0,485	0,024	0,002	0,043
stabilny	0,657	0,014	0,005	0,023
lepszy	0,402	0,005	0,003	0,000
grafika	0,327	0,022	0,009	0,006
prostota	0,780	0,044	0,025	0,073
przyzwyczajenie	0,953	0,152	0,012	0,555
grywalność	0,997	0,581	0,915	0,002

Źródło: Opracowanie własne.

Rysunek 1 przedstawia graficzną prezentację wyników analizy korespondencji dla całości badanej próby. Na podstawie danych zilustrowanych na rysunku 1 można stwierdzić, że pierwszy wymiar tłumaczy 63,3%, a drugi wymiar 24,7% całkowitej inercji. Grupa oprogramowania Open Source *gry* znajduje się najbardziej na lewo względem początku układu, tak samo jak kategoria zmiennej *powody: grywalność* (rys. 1). Oznacza to, że dla *gier* częściej niż przy założeniu niezależności między kolumnami i wierszami tabeli wielodzielczej ankietowani wymieniali powód *lubię w nią grać*. Grupa oprogramowania Open Source *gry* przez to odróżnia się od innych grup oprogramowania Open Source. Najbardziej w górę od początku układu położona jest grupa oprogramowania Open Source *biurowe* oraz kategoria zmiennej *powody: przyzwyczajenie*. Można na tej podsta-

wie wnioskować, że w przypadku grupy oprogramowania Open Source *biurowe* częściej niż przy założeniu niezależności między wierszami a kolumnami tabeli wielodzielczej ankietowani wymieniali powód *używam go z przyzwyczajenia*. Najbardziej w dół od początku układu znajduje się grupa oprogramowania Open Source *systemy* oraz kategoria zmiennej *powody: uczenie się*. Oznacza to, że ankietowani częściej niż przy założeniu niezależności kolumn i wierszy tablicy wielodzielczej korzystają z grupy oprogramowania Open Source *systemy* ze względu na możliwość dzięki niej nauki. Kategorie zmiennej *powody* znajdujące się najbliżej początku układu to *darmowy*, *lepszy* oraz *legalny*. Nie występuje dla nich istotnie różnicujący związek z którąkolwiek grupą oprogramowania Open Source. Blisko siebie położone, przez to podobne do siebie, są grupy oprogramowania Open Source: *biurowe* i *hobby* oraz kategorie zmiennej *powody: popularny* i *prostota*, a także *nauka*, *systemy* oraz *programistyczne2*, z kategoriami zmiennej *powody: wymagany* i *rozwój_wpływ*). Na podstawie rysunku 1 nie istnieją podstawy do odrzucenia hipotezy o różnicy w deklarowanych powodach użycia grup oprogramowania Open Source dla całości badanej próby.



Rysunek 1: Graficzna prezentacja wyników analizy korespondencji dla całości próby

Źródło: Opracowanie własne.

W tabeli 5 przedstawiono wyniki analizy korespondencji dla kategorii zmiennej *powody* zgodnych z teorią TAM. Inercja dla całości badanej próby (dla zmiennej *powody_TAM*) wyniosła 0,0791. Dwa wymiary tłumaczą 94,84% całkowitej inercji (czyli analiza korespondencji bardzo dobrze tłumaczy zależności między badanymi zmiennymi). Największy wkład w całkowitą inercję mają grupy oprogramowania Open Source: *gry* (36,4% inercji), *biurowe* (29,1% inercji) oraz *systemy* (17,5% inercji). Spośród kategorii zmiennej *powody_TAM* największy wkład w inercję mają kategorie *zgodność z filozofią oprogramowania* (SF), *zadowolenie* (S) oraz *zgodność z dotychczasowym doświadczeniem* (CPE). Największy wpływ

na pierwszy wymiar mają grupy oprogramowania Open Source: *biurowe* oraz *gry* a także kategorie zmiennej *powody_TAM*: *zgodność z dotychczasowym doświadczeniem* (CPE) i *zadowolenie* (S). Na drugi wymiar największy wpływ mają grupy oprogramowania Open Source: *gry* i *systemy* oraz kategorie zmiennej *powody_TAM*: *zadowolenie* (S) oraz *zgodność z filozofią oprogramowania* (SF).

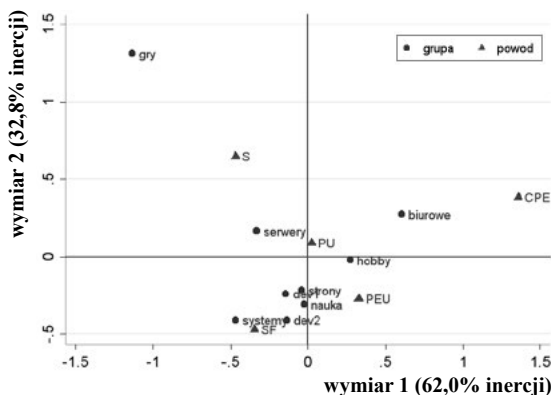
Tabela 5: Wyniki analizy korespondencji dla kategorii zmiennej powody zgodnych z teoria TAM

Kategorie	Ogólnie		Wymiar 1	Wymiar 2
	Jakość wytłumaczenia	% inercji wyjaśnionej	Wkład	Wkład
grupa				
biurowe	0,998	29,1	0,408	0,113
hobby	0,992	3,8	0,060	0,000
systemy	0,979	17,5	0,176	0,189
strony	0,593	1,3	0,001	0,023
serwery	0,984	3,4	0,046	0,016
gry	0,999	36,4	0,296	0,547
dev1	0,545	2,7	0,008	0,031
dev2	0,537	4,6	0,005	0,065
nauka	0,420	1,2	0,000	0,015
powody TAM				
PU	0,266	1,9	0,001	0,014
SF	0,965	19,8	0,131	0,333
CPE	0,983	38,6	0,578	0,064
PEU	0,824	11,9	0,105	0,100
S	0,988	27,8	0,185	0,490

Źródło: Opracowanie własne.

Rysunek 2 przedstawia graficzną prezentację wyników analizy korespondencji dla całości badanej próby (kategorie zmiennej *powody* zgodne z teorią TAM). Pierwszy wymiar tłumaczy 62,0%, a drugi wymiar 32,8% całkowitej inercji. Grupa *gry* znajduje się najbardziej na lewo względem początku układu, tak samo jak kategoria zmiennej *powody_TAM*: *zadowolenie* (S). Grupa oprogramowania Open Source *gry* najmocniej odróżnia się od grupy oprogramowania Open Source *systemy* położonej najbardziej w dół od początku układu. Najbardziej na prawo od początku układu znajduje się grupa oprogramowania Open Source *biurowe* oraz kategoria zmiennej *powody_TAM*: *zgodność z dotychczasowym doświadczeniem* (CPE). Kategoria zmiennej *powody_TAM* mająca większy niż przy założeniu niezależności między wierszami a kolumnami związek z grupą oprogramowania Open Source *systemy* to *zgodność z filozofią oprogramowania* (SF). Wynika z tego, że użytkownicy *systemów* operacyjnych Open Source częściej wymieniają jako powody używania tej grupy oprogramowania Open Source powody wynikające z treści licencji Open Source. Kategorią zmiennej *powody_TAM* znajdującą

się najbliższej początku układu jest *postrzegana przydatność* (PU) – nie występuje dla niej istotnie różnicujący związek z którąkolwiek grupą oprogramowania Open Source. Blisko siebie położone (przez to podobne do siebie) są grupy oprogramowania Open Source *systemy i programistyczne2* oraz (w tej samej ćwiartce układu) *programistyczne1*, *nauka* i *strony*.



Rysunek 2: Graficzna prezentacja wyników analizy korespondencji dla całości próby (powody zgodne z teorią TAM)

Źródło: Opracowanie własne.

5. Podsumowanie

Najczęściej wymienianym przez użytkowników powodem użycia oprogramowania Open Source okazuje się niski koszt tego oprogramowania. Bezpłatność oprogramowania Open Source i jego postrzegana przydatność dla użytkownika nie stanowi jednak specyficznego powodu użycia dla którejkolwiek grupy oprogramowania objętego tą licencją. Użytkownicy wybierają gry Open Source, ponieważ lubią w nie grać. W przypadku systemów operacyjnych częściej od innych grup oprogramowania Open Source powodem użycia jest możliwość dzięki nim nauki, również poprzez czytanie i recenzowanie udostępnionego kodu źródłowego aplikacji. Narzędzia biurowe i hobbystyczne Open Source wybierane są z powodu przyzwyczajenia lub ich popularności wśród znajomych i rodziny użytkownika. Analiza dla powodów zgodnych z teorią *Technology Acceptance Model* (TAM) wykazała, że użycie oprogramowania biurowego i hobbystycznego Open Source wynika z postrzeganej przez użytkownika zgodności obsługi tych programów z dotychczasowym doświadczeniem użytkownika, ponieważ interfejsy programów objętych tą licencją projektowane są zgodnie z „regulą najmniejszego zaskoczenia”. Powody użycia systemów operacyjnych, takich jak GNU/Linux, BSD, Open Solaris, przez użytkownika są zbieżne z treścią licencji Open Source, np. dostęp do kodu źródłowego, czy możliwość jego modyfikacji. Użycie gier Open Source wiąże z zadowoleniem z korzystania z nich.

Zastosowana dla potrzeb analizy metodologia była wzorowana na metodologiach wykorzystanych w dotychczasowej literaturze, nie była jednak pozbawiona wad. Badania preferencji oparte na danych jakościowych z reguły obciążone są mocną subiektywnością. Dodatkowo, należy ostrożnie podchodzić do wyciąganych wniosków. Analiza opierała się na *deklarowanych* powodach, więc nie można wykluczyć, że część ankietowanych wskazywała odpowiedzi, które według nich *wypadaloby* zaznaczyć, w miejsce *realnych* powodów. Co więcej, sama konstrukcja pytań w kwestionariuszu będących podstawą analizy jakościowej mogła prowokować do udzielania podobnych odpowiedzi niezależnie od grupy oprogramowania Open Source, do której odnosi się pytanie. Istnieje prawdopodobieństwo, że gdyby ankietowani musieli samodzielnie wpisać czynniki motywujące, odpowiedzi byłyby bardziej zróżnicowane. Niestety, wymóg samodzielnego wpisania mógłby działać zniechęcająco, obniżając liczbę całkowicie wypełnionych kwestionariuszy, co przy wybranej technice analizy wymagającej dużych zbiorów danych mogłoby okazać się poważniejszym problemem. Ponadto, samodzielniewypełnianie odpowiedzi również nie gwarantuje, że ankietowany odpowie szczerze.

Otrzymane wyniki dotyczące postrzeganej przez użytkownika przydatności oprogramowania Open Source są zgodne z wynikami otrzymanymi przez Gwembu i Wang. Popularność niskiego kosztu oprogramowania Open Source jako powodu jego używania jest zgodna ze spostrzeżeniem Allmana.

Uzyskane wyniki jednoznacznie pokazują, że użycie oprogramowania Open Source przez użytkowników nie wynika jedynie z możliwości nabycia go za darmo. Chociaż powód użycia tego oprogramowania związany z „ekonomią prezentu”, czyli niski koszt programów był najczęściej wymienianym przez ankietowanych, w niniejszym artykule zostało dowiedzione, że fakt korzystania z Open Source może być również związany z innymi powodami w zależności od grupy oprogramowania. Do tych powodów należą np. możliwość nauki dzięki użyciu programu, przyzwyczajenie użytkownika, czy prostota obsługi programu.

Bibliografia

[1] Allman E. Sendmail. [w:] A. Brown, G. Wilson, The architecture of Open Source applications, Volume II: Structure, Scale, and a few more fearless hacks. [dostęp online 16 marca 2013] <http://aosabook.org>.

[2] Bonaccorsi A., Rossi C., (2006). Comparing motivations of Individual Programmers and firms to take part in the Open Source Movement. *Knowledge, Technology and Policy* nr 18.

[3] Crowston K., Wei K., Howison J. (2012). Free/Libre Open Source Software Development: What we know and what we do not know. *ACM Computing Surveys*, vol. 44 nr 2.

- [4] Gacek C., Arief B., (2004). The Many Meanings of Open Source. IEEE Software. *IEEE Computer Society*, vol 21.
- [5] Gallego M.D., Luna P., Bueno S., (2008). User acceptance model of open Source software, *Computers in Human Behaviour* nr 24.
- [6] Gagné M., (2005). Jak to się robi w Linuksie?. Helion, Gliwice.
- [7] Gwembu K.L., Wang J., (2010). Seeing eye to eye? An explanatory study of free open Source software users' perceptions. *The Journal of Systems and Software* nr 83.
- [8] Hann I., (2004). Why developers participate in Open Source software projects: An empirical investigation. *Twenty-Fifth International Conference on Information Systems*.
- [9] Hars A., (2002). Working for free? – Motivations of participating in Open Source projects. *International Journal of Electronic Commerce*, vol 6 nr 3.
- [10] Koski E., (2005). OSS production and licensing strategies of software firms. *Review of Economic Research on Copyright Issues*, vol 2 nr 2, s. 111–125.
- [11] Lerner J., Tirole J., (2005). The Scope of Open Source Licensing. *Journal of Law, Economics and Organization*, vol 21, s. 20–56.
- [12] Lerner J., Tirole J., (2005). The Economics of Technology Sharing: Open Source and Beyond. *Journal of Economic Perspectives*, vol 19 nr 2.
- [13] Raymond E.S., (2004). UNIX. Sztuka programowania. Helion, Gliwice.
- [14] Raymond E.S. Homesteading the Noosphere [dostęp online 4 grudnia 2012] <http://catb.org/esr/writings/homesteading/homesteading>.
- [15] van Rossum G. Open Source Summit Trip Report. [dostęp online 18 marca 2013] <http://linuxgazette.net/issue28/rossum.html>
- [16] Schiff A., (2002). The Economics of Open Source Software: A Survey of the Early Literature. *Review of Network Economics*, vol 1.
- [17] Sen R., (2006). Open Source Software Development Projects: Determinants of Project Popularity. *EERI Research Paper Series* nr 2.
- [18] Sokół R., (2006). Slackware Linux. Helion, Gliwice.
- [19] Stanimir A., (2005). Analiza korespondencji jako narzędzie do badania zjawisk ekonomicznych. Wydawnictwo Akademii Ekonomicznej we Wrocławiu, Wrocław.
- [20] Pentor Research International, (2010). Wykorzystanie Wolnego i Otwartego Oprogramowania w rządowej administracji publicznej. Poznań. [dostęp online 11 maja 2013] http://pppit.org.pl/publikacje/badanie_pentor.pdf
- [21] http://osnews.com/story/10142/The_Big_Arch_Linux_Interview/ [dostęp online 18 marca 2013]
- [22] <http://perens.com/works/articles/Economic.html> [dostęp online 18 grudnia 2012]

