

Magdalena Szpunar

**INTERNET A KULTURA DARU.
FENOMEN RUCHU *OPEN SOURCE***

Powstanie pierwszego cyfrowego komputera ENIAC, zbudowanego w roku 1946 w USA, zainicjowało powstanie grupy entuzjastów programowania komputerowego. Osób, które nie tylko tworzą oprogramowanie, ale także się nim bawią. Nowa technologia skupiła wokół siebie głównie ludzi młodych, którzy pisząc programy komputerowe, mniej lub bardziej świadomie przyczynili się do wytworzenia własnej kultury¹. Tworzyli ją hakerzy, czyli osoby biegłe w zagadnieniach technicznych, którzy znajdowali upodobanie w rozwiązywaniu problemów i przekraczaniu ograniczeń², nie tylko oprogramowania, ale swojego umysłu.

Najwyższą wartością w kulturze hakerów jest usprawnianie funkcjonującego oprogramowania, bądź tworzenie nowego. Miejsce i znaczenie w hierarchii hakerskiej zależy od wkładu w rozwój oprogramowania, a wkład ten oceniany jest przez członków społeczności³. Zasadne jest udzielenie odpowiedzi na pytanie, dlaczego tysiące ludzi dobrowolnie, nie otrzymując w zamian żadnego wynagrodzenia, poświęca swój wolny czas, życie rodzinne, by pracować bezinteresownie dla idei *open source*? Działania podejmowane przez hakerów nie przystają do silnie zmaterializowanej i konsumpcyjnej kultury współczesnej, w której niemal wszelkie transakcje opierają się o wymierne korzyści materialne. We współczesnym świecie zdominowanym przez pieniądze,

¹ C. Meszyński, *Wpływ kultury hakerskiej na rozwój społeczeństwa informacyjnego. Hakerzy jako awangarda technologiczna*, [w:] L. Haber (red.) *Spółeczeństwo informacyjne, Aspekty funkcjonalne i dysfunkcjonalne*, Wyd. UJ, Kraków 2006, s. 149.

² Ibidem, s. 150.

³ M. Castells, *Galaktyka Internetu*, Rebis, Poznań 2003, s. 50.

takie wartości, jak altruizm ulegają deprecjacji, niemal całkowicie się dezaktualizując. Wydaje się, że trudno w tym świecie wskazać sferę życia, której nie można zmaterializować, niemal wszystko ma swoją cenę. Coraz częściej inne osoby traktowane są instrumentalnie. Pomoc innym jest traktowana raczej w kategoriach interesowności (osoba, której pomogłem może mi się kiedyś przydać), niż w aspekcie bezinteresownej pomocy, nieoczekującej niczego w zamian. Już w społeczeństwach pierwotnych jednostki wymieniały ze sobą nie tylko dobra i bogactwa, ale przede wszystkim uprzejmości, festyny, obrzędy, tańce, czy święta⁴. W kulturze tej status społeczny jednostek wyznaczała bezinteresowna wymiana dóbr. Wymiana społeczna jak zauważa Peter Blau dokonuje się wówczas, gdy ktoś daje pieniądze biedakowi, ponieważ liczy na wyrazy wdzięczności i szacunek z ich strony⁵. Choć darczyńca nie otrzymuje bezpośredniej „zapłaty” od obdarowanego, to ten obdarza go swoim szacunkiem i wdzięcznością. Warto podkreślić, że w przypadku wymiany społecznej zobowiązania nie są określone, zaś w przypadku wymiany ekonomicznej, która z reguły opiera się o formalne umowy, mamy dokładnie określone, jakie wielkości należy wymienić⁶. Bronisław Malinowski posługuje się tutaj zasadą wzajemności *do ut des* – daję abyś dał, a Marcel Mauss wskazuje, że w społeczeństwach pierwotnych zobowiązanie obdarowanego do ofiarowania daru, stanowiło sieć spajającą społeczeństwo⁷. Dar tworzył więź między dającym i biorącym, tworząc solidarność społeczną, a nie tylko – jak to ma miejsce w przypadku handlu – wymianę usług, czy przedmiotów⁸. Zobowiązanie obdarowanego wobec darczyńcy Zbigniew Nęcki określa mianem emocjonalnego zadłużenia wobec nadawcy⁹. Również i w chrześcijaństwie mamy do czynienia z egzemplifikacją zasady *do ut des*. Chrześcijanie bowiem wierzą, iż bezinteresowna dobroć zostanie odwzajemniona, powracając do wiernego prawu chrześcijanina, np. nagroda za altruizm jest pewna (przysłowiowy wdowi grosz), ale czekać na nią można aż do przyszłego życia¹⁰.

⁴ M. Mauss, *Socjologia i antropologia*, PWN, Warszawa 1973, s. 217.

⁵ P. Blau, *Wymiana społeczna*, [w:] A. Jasińska-Kania, L. Fijałkowski, J. Szacki, M. Ziółkowski (red.) *Współczesne teorie socjologiczne*, Scholar, Warszawa 2006, s. 84.

⁶ Ibidem, s. 85.

⁷ E. Nowicka, *Świat człowieka-świat kultury*, Warszawa, PWN, 2007, s. 294.

⁸ Ibidem, s. 295.

⁹ Z. Nęcki, *Atrakcyjność wzajemna*, Kraków, Wydawnictwo Profesjonalnej Szkoły Biznesu, 1996, s. 52.

¹⁰ J. Bierówka, *Zasada wzajemności w formującym się społeczeństwie informacyjnym*, Kraków, 2006 – niepublikowana praca doktorska, s. 17.

M. Sahlins wyróżnił trzy podstawowe typy wzajemności. *Wzajemność uogólniona* (solidarność) stanowi podstawę wymian altruistycznych, gdzie strony wymiany oferują sobie wzajemną pomoc. *Wzajemność uogólniona* przypomina *wymianę autyczną*, która oznacza dawanie jednostronnych prezentów, bez oczekiwania nagrody w postaci pewnego zachowania odbiorcy lub osób trzecich¹¹. Jej bardziej „egoistyczną wersją” jest tzw. słaba wzajemność, która nie określa ściśle wzajemnych zobowiązań, czego przykładem jest zdaniem Sahlinsa hojność, pomoc, gościnność, a także powinności rodzinne, powinność wodza i zasada *noblesse oblige* (szlachectwo zobowiązuje). *Wzajemność zrównowazona* mówi, iż wymieniane dobra muszą być ekwiwalentne, a wymiana zachodzi bezpośrednio. Egzemplifikacją tego typu wzajemności są umowy przyjacielskie, wymiany darów, czy traktaty pokojowe. *Wzajemność negatywna* określa sytuacje, w których jedna ze stron uzyskuje jakieś dobro, ale nie domaga się rewanżu, czego przykładem może być targowanie się, hazard, ale także oszustwo i kradzież¹² (rys. 1).



Rys. 1. Typy wzajemności według M. Sahlinsa

Źródło: opracowanie własne, na podstawie: M. Sahlins, *Socjologia wymiany w społeczeństwach pierwotnych*. w: M. Kempy, J. Szmatka (red.), *Współczesne teorie wymiany społecznej. Zbiór tekstów*. Wydawnictwo Naukowe PWN, Warszawa, 1992.

Początki ruchu *open source* sięgają roku 1984, kiedy to Richard Stallman, programista MIT¹³, powołał do życia *Free Software Foundation*, gdzie proponował, by prawo autorskie (*copyright*) zostało zastąpione zasadą *copyleft*, wedle której każdy, kto korzysta z wolnego oprogramowania, powinien udostępniać w zamian wszelkie jego usprawnienia w Internecie. Propozycja Stallmana była reakcją na roszczenia *American Telephone & Telegraph* do praw autorskich systemu Unix. Stallman

¹¹ P. Blau, *Wymiana społeczna*, op.cit., s. 84.

¹² M. Sahlins, *Socjologia wymiany w społeczeństwach pierwotnych*. [w:] M. Kempy, J. Szmatka (red.), *Współczesne teorie wymiany społecznej. Zbiór tekstów*, PWN, Warszawa 1992, s. 140–141.

¹³ MIT – *Massachusetts Institute of Technology* to jedna z najbardziej prestiżowych uczelni technicznych świata. Badania prowadzone w MIT zapoczątkowały wiele nowych gałęzi techniki m.in. biotechnologię, czy nanotechnologię.

stworzył system operacyjny GNU (*Gnu's Not Unix*), umieszczając go w Internecie i umożliwiając swobodne korzystanie z niego, pod warunkiem przestrzegania klauzuli *copyleft*¹⁴. System GNU dzięki Stallmanowi stał się pośrednio narzędziem walki o wolność słowa i wolnego korzystania z wolnego oprogramowania, co zdaniem założyciela FSF (*Free Software Foundation*) jest fundamentalnym prawem jednostki. Idee Stallmana wykorzystał Linus Torvalds, który opracował wersję Unixa nazwaną Linuxem. Rozpowszechniał ją za darmo w Internecie, prosząc użytkowników, by ją dalej rozwijali i udostępniali innym jej ulepszone wersje¹⁵. Obecnie Linux to szósty na świecie system operacyjny (0,8% udziału w rynku), a wyprzedzają go takie systemy jak Windows XP (71,2%), Windows Vista (16,14%), MacIntel (5,25%), Mac OS (2,69%) i Windows 2000 (2,11%)¹⁶. Mimo relatywnie niskiego udziału w rynku odbiorców indywidualnych, Linux jest głównie stosowany w superkomputerach, czy serwerach www. Według rankingu TOP500 w czerwcu 2008 roku, Linux był stosowany w $\frac{3}{4}$ największych maszyn obliczeniowych na świecie, podczas gdy udział systemu MS Windows wynosił zaledwie 1%¹⁷. Powyższe dane dobitnie świadczą o wysokiej jakości systemu Linux. Obecny kształt sieci WWW w znaczący sposób opiera się o oprogramowanie *open source*, ponieważ zarówno serwery, jak i serwisy WWW oraz bazy danych, funkcjonują dzięki niemu. Do najpopularniejszych projektów *open source* należą serwer Apache, język programowania witryn WWW – PHP, baza danych MySQL i PostgreSQL.

W ostatnim czasie flagowym produktem OS jest przeglądarka Firefox. W Polsce ponad 40% użytkowników Internetu korzysta z przeglądarki Firefox, a liczba jej użytkowników stale rośnie. *Open source*'owy Firefox w opinii specjalistów, jak i zwykłych użytkowników, jest programem o wiele lepszym, niż przeglądarka Internet Explorer firmy Microsoft dołączana do systemu MS Windows.

Eric Raymond oprogramowanie *open source* definiuje jako swobodnie rozpowszechniane, dające łatwo się rozwijać i modyfikować w celu przystosowania go do zmieniających się okoliczności¹⁸.

¹⁴ M. Castells, *Galaktyka Internetu*, op.cit., s. 24.

¹⁵ Ibidem, s. 25.

¹⁶ <http://marketshare.hitslink.com/report.aspx?qprid=10&qpmr=24&qpdt=1&qpct=3&qpcal=1&qptimeframe=M&qpsp=113>.

¹⁷ <http://www.top500.org/stats/list/31/os>.

¹⁸ E. Raymond, *Zagospodarowywanie noosfery*, tł. A. Skura, http://pl.wikisource.org/wiki/Zagospodarowanie_noosfery.

Pisząc o ruchu *open source* należy dokonać koniecznego różnienia na ruch wolnego oprogramowania (*free software*) i otwartego oprogramowania (*open source*), choć zasadnicze idee obu ruchów są niemal takie same, a nierzadko wręcz tożsame, różnią się między sobą niuansami w filozofii działania i zwróceniem uwagi na etyczne aspekty wolnego oprogramowania. Z ruchem *open source* związane są nazwiska takie jak Eric Raymond, Linus Torvalds, John Hall, Larry Augustyn i Bruce Perens, natomiast z ruchem wolnego oprogramowania (*free software*) wiąże się zasadniczo tylko jedno nazwisko – R. Stallmana. Założyciele ruchu *open source* zwracają uwagę, iż promując wolne oprogramowanie, należy zwracać uwagę głównie na aspekty jakości tworzonego oprogramowania. Utożsamiany z ruchem *open source* E. Raymond w słynnym eseju *The Cathedral and the Bazaar*, przedstawia dwa modele tworzenia FLOSS (*Free Libre/Open Source Software*), tj. *model katedry*, w którym kod źródłowy jest udostępniany z każdą nową wersją programu, ale kod zna i tworzy jedynie wąska grupa specjalistów oraz *model bazaru*, gdzie kod tworzy się w Internecie przy współudziale chętnych¹⁹. Sam Raymond początkowo był zwolennikiem modelu katedry, gdyż programy „powinny powstawać jak katedry: budowane zręcznymi palcami samotnych czarodziei lub grupy dostojnych magów pracujących w pełnym namaszczeniu w odosobnieniu”²⁰. Szybko jednak autor *Katedry i Bazaru* przekonał się, że zbiorowa mądrość i inteligencja tłumów gwarantuje sukces danego projektu programistycznego.

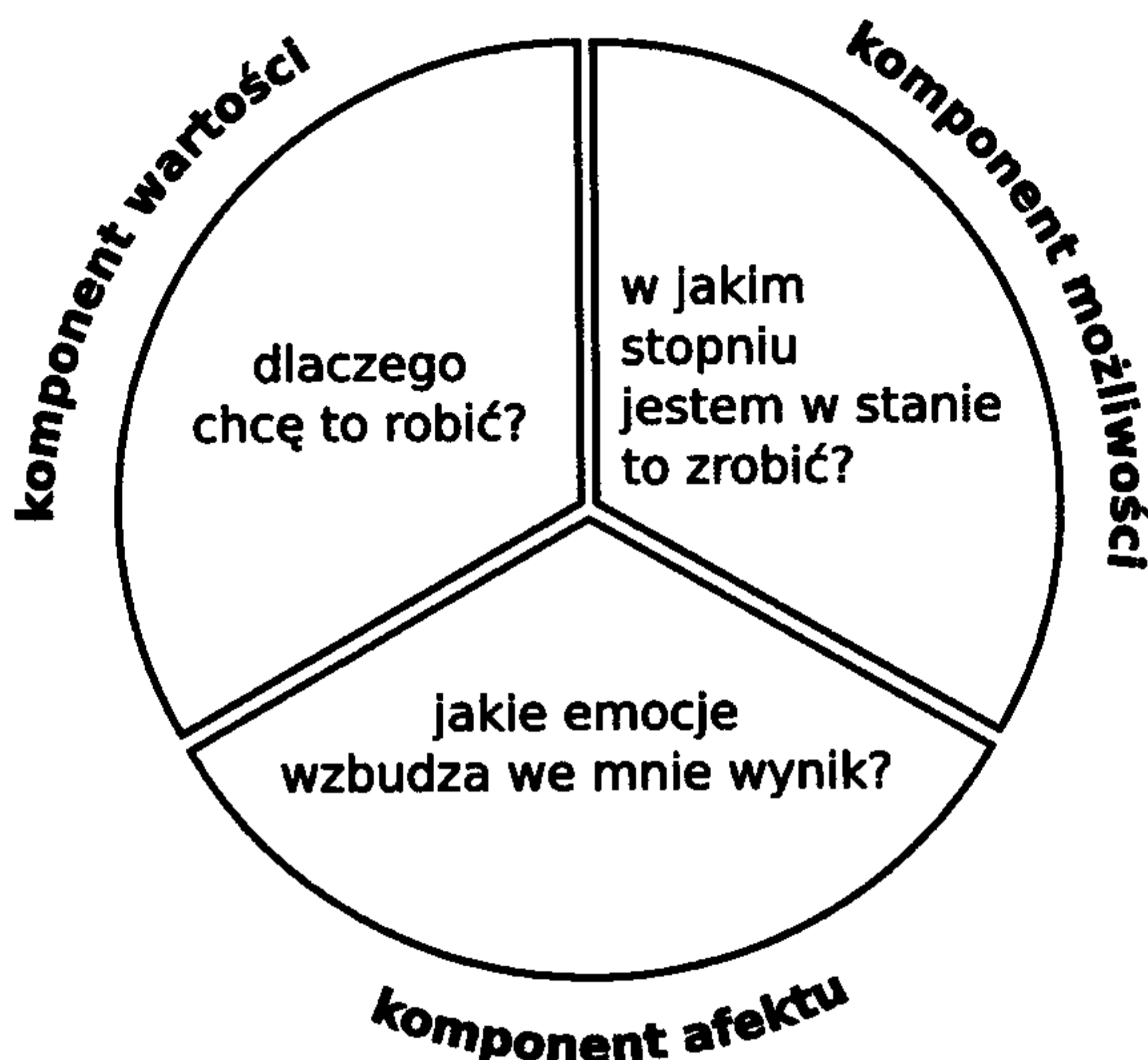
Analogicznie jak w każdym procesie motywacyjnym programiści *open source* muszą sobie odpowiedzieć na trzy pytania: dlaczego chcę to robić? (*komponent wartości*), w jakim stopniu jestem w stanie to zrobić? (*komponent możliwości*) oraz jakie emocje wzbudza we mnie założony wynik i zakładane działanie? (*komponent afektu*)²¹. Tendencja do podjęcia jakiegokolwiek działania, jest tym silniejsza, im większa jest szacowana wartość celu²². Programista *open source* podejmie działanie, jeśli wie, że jego efekt będzie dla niego korzystny np. znajdzie lepszą pracę, czy zdobędzie uznanie w środowiska. Oczywiście cel nierealny, czy braki umiejętności (*komponent możliwości*) wyklucza osiągnięcie sukcesu w tym zakresie (rys. 2).

¹⁹ http://pl.wikipedia.org/wiki/The_Cathedral_and_the_Bazaar.

²⁰ E. Raymond, *Katedra i Bazar*, tłum. A. Skura, 2001, <http://www.linuxcommunity.pl/node/4>

²¹ W. Łukaszewski, D. Doliński, *Mechanizmy leżące u podstaw motywacji*, [w:] J. Strelau (red.) *Psychologia. Podręcznik akademicki*, GWP, Gdańsk 2006, t. 2., s. 451.

²² Ibidem, s. 450.



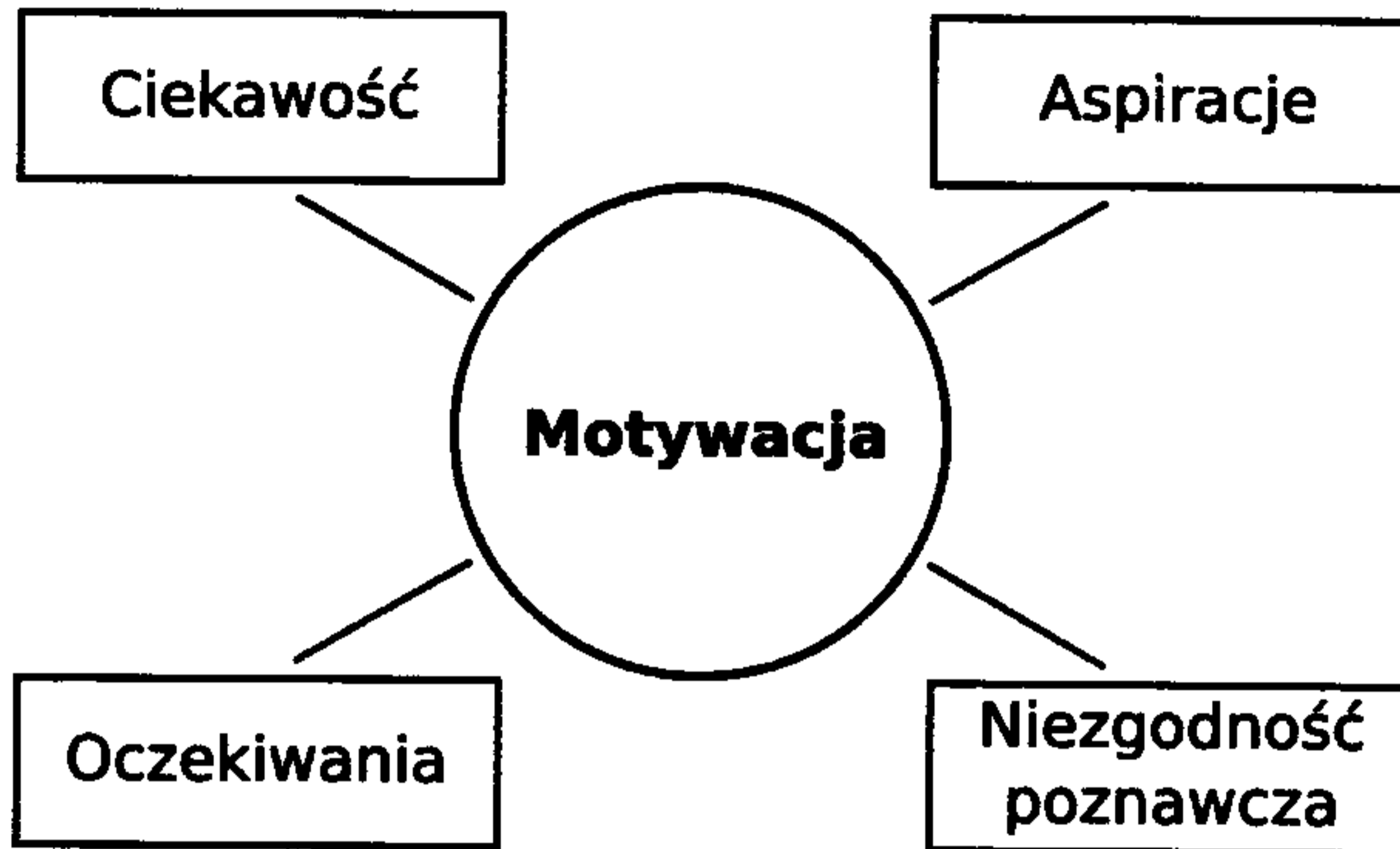
Rys. 2. Komponenty składowe motywacji przy podejmowaniu działalności *open source*

Źródło: opracowanie własne, na podstawie: W. Łukaszewski, D. Doliński, *Mechanizmy leżące u podstaw motywacji*, w: J. Strelau (red.) *Psychologia. Podręcznik akademicki*, Gdańsk GWP, 2006, t. 2

Jak dowodzą psychologowie mechanizmy motywacyjne wiążą się z ciekawością poznawczą, oczekiwaniami, aspiracjami, fantazjami i marzeniami oraz różnymi formami niezgodności poznawczej²³. Sukcesy intelektualne, jak dowodził Berlyne wiążą się nie tyle ze zdolnościami, ale *ciekawością* i dążeniem do jej zaspokojenia. Istotnym czynnikiem w procesie motywacji są *oczekiwania*, które mogą mieć charakter losowy, gdzie wynik działania nie jest zależny od jednostki lub sprawnościowy, gdzie efekty zależą od działania jednostki. Badania prowadzone m.in. przez Lewina i Oettingena dowodzą, że im większe są oczekiwania, tym większa jest motywacja, a w konsekwencji większa satysfakcja z osiągniętego wyniku działania. Okazuje się również, że nawet gdy oczekiwania są niezbyt realistyczne, silniej angażują one jednostkę i prowadzą do lepszego rezultatu. Jednostkę bardzo silnie do działania stymulują również posiadane *aspiracje*. Czasem mogą one przybierać formę wyłącznie marzeń, co prowadzi do sytuacji, gdzie jednostka nie podejmuje działań, a skupia się jedynie na wyobrażeniu stanu pożą-

²³ W. Łukaszewski, D. Doliński, op.cit., s. 445–446.

danego. Motywatorem mogą być również różne formy *niezgodności poznawczej*, np. gdy opinie innych nie zgadzają się z tym, co myślimy o sobie lub gdy opinie różnych ludzi na nasz temat są sprzeczne²⁴.



Rys. 3. Główne mechanizmy motywacyjne

Źródło: opracowanie własne, na podstawie: W. Łukaszewski, D. Doliński, *Mechanizmy leżące u podstaw motywacji*, w: J. Strelau (red.) *Psychologia. Podręcznik akademicki*, Gdańsk GWP, 2006, t. 2

Wszystkie wymienione motywatory mają kluczowe znaczenie w działaniach podejmowanych przez programistów *open source*. Ciekawość poznawcza ma kluczowe znaczenie w sytuacji, gdy twórca *open source* ma do czynienia z zadaniem trudnym, wymagającym sporego zaangażowania, takim, z którym do czynienia ma po raz pierwszy i niełatwo określić jego efekt końcowy. Wolontariusze *open source* mogą być motywowani również oczekiwaniami, które w ich przypadku mają głównie charakter sprawnościowy. To od ich wkładu pracy zależy ocena środowiska, prestiż i szacunek społeczności. Oczekiwania pozwalają im domniemywać, że działalność *open source* może polepszyć ich pozycję na rynku pracy, a przez to może stać się przepustką do lepiej płatnej pracy. Skoro ponoszą koszt w postaci wkładu własnej pracy, nierzadko zaniedbując inne czynności życiowe, mają prawo oczekiwać, że ich działanie przyniesie kiedyś jakiś rezultat, choć jest to oczywiście forma gratyfikacji odroczonej. W wielu przypadkach to wysokie aspiracje powodują, że informatycy podejmują się udziału w projektach *open source*'owych. Zadania, które powierzono im w ich firmach są

²⁴ Ibidem, s. 445–450.

dla nich za mało ambitne, nie mają więc szans na pokazanie swoich umiejętności, a przez to nierzadko nie rozwijają się, a wręcz następuje regres ich umiejętności zawodowych. Aspiracje sprawiają, że poza pracą *stricte* zawodową szukają nowych wyzwań, które pozwalają w pełni im się rozwijać. Wydaje się, że aspiracje osób podejmujących tego typu działania są szczególnie wysokie. Informatycy wybierają działania open source'owe gdy chcą udowodnić innym, a często również sobie, że mają wysokie kwalifikacje. Dzięki tej działalności mają możliwość sygnalizowania swoich wysokich umiejętności. W sytuacji, gdy są niedoceniani, są źle oceniani mimo dużego wkładu pracy, „uciekają” w sferę, która pozwala im w pełni rozwijać skrzydła. „Ja im jeszcze udowodnię” – te słowa w wielu przypadkach stają się silnym stymulatorem działania, próbując niwelować niezgodność poznawczą pomiędzy opiniami innych, a przekonaniami własnymi. W sytuacji gdy czują się jedynie niewielkim trybikiem w maszynie, mają mały zakres autonomii, czy nie wykorzystuje się ich umiejętności, szukają nowych dróg realizacji swoich pasji programistycznych.

Głównym motywatorem działania programistów *open source* jest prestiż, reputacja i szacunek, jakie uzyskują w społeczności, dzięki swojej pracy. Zgodnie z tym, co głosi teoria wymiany społecznej: „zachowania społeczne są wymianą dóbr, zarówno materialnych, jak niematerialnych, takich jak symbole aprobaty i prestiżu”²⁵, uznanie zdobywa się nie tylko dzięki temu, że coś się podarowało, ale również, dlatego, że stworzyło się coś cennego²⁶. Nie istotne są w tej kulturze główne czynniki stratyfikacji znane nam ze społeczeństw współczesnych, a więc wykształcenie, zawód, czy osiągnięty dochód. Haker musi się wykazać dużym wkładem własnej pracy, inwencją twórczą, by efekty jego pracy zostały docenione. Wyrazy aprobaty i sympatii, czyli bodźce niematerialne w wielu przypadkach stanowią istotny czynnik motywacji programistów *open source*. P. Blau podkreśla, że „konkretne wymieniane korzyści są przede wszystkim cenione jako symbole poparcia i przyjaźni”²⁷. W społeczności hakerskiej większość członków zna jedynie swoje nicki (internetowe przezwiska), a nie imiona i nazwiska, którymi posługują się w realnym świecie. Istotnym elementem działania programistów *open source* jest również satysfakcja z wy-

²⁵ G. Homans, *Zachowania społeczne jako wymiana dóbr*, [w:] W. Derczyński, A. Jasińska-Kania, J. Szacki (red.) *Elementy teorii socjologicznych*, PWN, Warszawa 1975, s. 119.

²⁶ M. Castells, *Galaktyka Internetu*, op.cit., s. 59.

²⁷ P. Blau, *Wymiana społeczna*, op.cit., s. 86.

konywanej pracy. Jak zauważa Manuel Castells hakerom radość daje sam akt tworzenia, analogicznie jak to ma miejsce w przypadku pędu twórczego artystów. Jednostki bowiem działają pod wpływem dwóch stanów metamotywacyjnych: telicznego i paratelicznego. W stanie *telicznym* jednostka ukierunkowana jest na realizację osiągnięcia celu, zaś w stanie *partelicznym* jednostka motywowana jest nastawieniem na samą aktywność, a nie jej instrumentalny charakter. Parateliczne działanie jest dla jednostki istotne nie ze względu na to, że może ono doprowadzić do jakiegoś rezultatu, ale dlatego, że samo w sobie jest satysfakcjonujące i angażujące. Działanie takie charakterystyczne jest dla artystów, kompozytorów, czy poetów, którzy podejmując działanie, zapominają o świecie zewnętrznym²⁸. Motywacja parateliczna cechować będzie również wielu programistów *open source*, dla których motywatorem działania jest prawdziwa radość tworzenia i rozwiązywanie problemów informatycznych. Nierzadko zapominają o spaniu czy jedzeniu dopóki nie rozwiążą konkretnego problemu. Jak zauważa E. Raymond „Szczęśliwy programista to taki, który nie jest ani niewykorzystywany, a nieprzytłoczony ciężarem źle sformułowanych celów i stresującego tarcia tworzącego się w procesie tworzenia. Radość źródłem wydajności”²⁹. Im większe przekonanie do tego, co się robi, satysfakcja z wykonywanej pracy, tym większy zapał do pracy i lepsze efekty się uzyskuje. Co warte podkreślenia wkład programistyczny jest oceniany jako znaczący jedynie wtedy, gdy nie służy wyłącznie własnemu dobru³⁰. Społeczną pozycję hakera nie tworzy dominacja, wdzięk osobisty, czy zdolność promowania się, ale ofiarowywanie innym tego, co ma się najcenniejszego – czasu i umiejętności. By zdobyć szacunek innych rezygnują z wielu przyjemności czasu wolnego, a nawet życia rodzinnego. Twórca największego projektu *Open Source* Linus Torvalds nad jądrem systemu operacyjnego Linux pracował bez przerwy pięć miesięcy³¹.

Kultura daru, jak podkreśla Ignacy Fiut, oparta jest o idee wolności i darmowego przekazywania oprogramowania jego użytkownikom. Funkcjonuje w sposób analogiczny do kultury akademickiej, w której tradycją jest wspólne uprawianie pracy naukowej, poddawanie się oce-

²⁸ W. Łukaszewski, D. Doliński, *Mechanizmy leżące u podstaw motywacji*, op.cit., s. 446.

²⁹ E. Raymond, *Katedra i Bazar*, op.cit.

³⁰ M. Castells, *Galaktyka Internetu*, op.cit., s. 61.

³¹ C. Meszyński, *Wpływ kultury hakerskiej na rozwój społeczeństwa informacyjnego. Hakerzy jako awangarda technologiczna*, op.cit., s. 152.

nie środowiska, udostępnianie wyników swoich badań³². Naukowcy nieustannie poddają weryfikacji wyniki swoich badań, a ocena środowiska staje się jednym z głównych czynników rozwoju i stymulatorów dalszej pracy. Konstruktywna krytyka w wielu przypadkach jest niezbędna, pozwala odkryć problemy w nowym kontekście lub dostrzec pomijane nisze. Z analogiczną sytuacją mamy do czynienia w kulturze hakerskiej. Możliwość opiniowania powszechnie dostępnego kodu źródłowego, staje się sposobem na osiągnięcie kodu wysokiej jakości i niezawodnego³³. Współtwórcy systemu Linux sami podejmują nowe inicjatywy, omawiają na forach problemy programistyczne, a członkowie społeczności skupionej wokół Linuxa pomagają w procesie jego testowania i ulepszania³⁴. Jak wskazuje M. Castells mamy tutaj do czynienia z procesem uczenia się w działaniu, gdzie odkrycia i zastosowania oddziałują na siebie wzajemnie i są sprawdzane w procesie prób i błędów³⁵. Pomoc innych w wielu przypadkach gwarantuje rozwiązywanie karkołomnych algorytmów programistycznych, z którymi jedna osoba nie bardzo jest w stanie sobie poradzić. Dopiero „zbiorowa mądrość” pozwala uzyskać efekt lepszy, od indywidualnie wykonywanej pracy. Egzemplifikacją tego typu działań jest system Linux, którego sukces przypisywać można wkładowi tysięcy użytkowników, wprowadzających nowe rozwiązania, zamieszczających w Internecie wolne od opłat poprawki, odwzajemniając techniczny dar, który otrzymali na początku³⁶. Przykładem jest stale udoskonalanie systemu operacyjnego Linux przez tysiące użytkowników i programistów, dzięki czemu system uznawany jest za najbardziej zaawansowany na świecie³⁷. Jak słusznie zauważa bowiem M. Castells „tylko sieć tysięcy mózgów podejmująca wspólny wysiłek, spontanicznie dzieląca się pracą, luźne acz skuteczne koordynowana, może zrealizować wyjątkowe zadania stworzenia systemu operacyjnego, który poradzi sobie z coraz bardziej skomplikowanymi, dysponującymi coraz większą mocą obliczeniową komputerami podłączonymi w sieć”³⁸.

³² I.S. Fiut, *Komunikacja sieciowa źródłem interakcji społecznych*, w: L. Haber (red.) *Spoleczeństwo informacyjne. Wizja czy rzeczywistość?* Kraków, Wyd. AGH, 2004, s. 313.

³³ E. Raymond, *The Cathedral and The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, O'Reilly, Sewastopol, 1999 – podają za: M. Castells, *Galaktyka Internetu*, Poznań, Rebis, 2003, s. 58.

³⁴ M. Castells, *Galaktyka Internetu*, op.cit., s. 60.

³⁵ M. Castells, *Spoleczeństwo sieci*, PWN, Warszawa 2007, s. 75.

³⁶ Ibidem, s. 361.

³⁷ M. Castells, *Galaktyka Internetu*, op.cit., s. 25.

³⁸ Ibidem, s. 57.

Raymond twierdzi wręcz, że jedni znajdują problemy, a inni wiedzą jak je rozwiązać, nierzadko nawet samo znalezienie błędu jest trudniejsze, niż jego korygowanie. Każdy programista podchodzi inaczej do danego problemu, korzysta z innych narzędzi analitycznych, co zwiększa prawdopodobieństwo, że czyjaś optyka pozwoli dany problem rozwiązać³⁹.

Niestety niektórzy użytkownicy, nieznający idei *open source*, domagają się ulepszeń w programach, a ich żądania, nierzadko, mają charakter kategoryczny. Taka postawa spotyka się z ostrym sprzeciwem „starszyzny” *open source*, która przypomina, iż działania OS wykonywane są nieodpłatnie, a każdy, jeśli ma tylko ochotę, może ulepszyć produkt końcowy, jeśli tylko potrafi. Krytykowanie, a właściwie krytykanctwo, które nic nie wnosi poza dyskredytowaniem pracy innych, spotyka się z falą sprzeciwu.

Wymiana społeczna jest jak zauważa P. Blau, stanem pośrednim pomiędzy kalkulacją korzyści, a czystym wyrazem miłości. Podobnie postępują programiści *open source*. Wiedzą, że nie są w stanie sami wykonać jakiegoś projektu, wchodzą więc w stosunki wymiany z innymi programistami, a każdy z nich wymienia w sposób pośredni swoją wiedzę i umiejętności z innymi, w zamian otrzymując reputację, szacunek i uznanie. Co ciekawe, jak podkreśla E. Raymond, im bardziej programista umniejsza swój wkład w rozwój danego oprogramowania, tym częściej ludzie uznają, iż wszelkie innowacje są jego dziełem, pomniejszając swój osobisty wkład w rozwój danego programu⁴⁰. Kultura hakerska gardzi wręcz motywacją ego jako głównym stymulatorem działalności *open source*, a promowanie własnej osoby spotyka się z bezlitosną krytyką środowiska⁴¹. Wręcz, jak zauważa E. Raymond, „wielcy i starsi w plemienu kultury hakerskiej muszą łagodnie się wyrażać i za każdym razem z humorem pomniejszać swoją rolę, by nie utracić swojego statusu”⁴². O ile motywatory takie jak reputacja, profesjonalizm, prestiż są akceptowalne, o tyle osoby rozbuchanym ego, podkreślające swoje duże umiejętności, gardząc przy tym innymi, są zupełnie w tym środowisku nieakceptowane. Skromność i docenienie wkładu społeczności w rozwój danego programu gwarantuje sukces, społeczność bowiem pozwala dostrzec możliwe kierunki rozwoju, dba o dobre imię współtworzonego projektu, a w konsekwencji decyduje o jego być czy nie być. Ma to ogromne znaczenie, jeśli uświadomimy sobie fakt, iż aż 60–75% pro-

³⁹ E. Raymond, *Katedra i Bazar*, op.cit.

⁴⁰ E. Raymond, *Katedra i Bazar*, op.cit.

⁴¹ E. Raymond, *Zagospodarowywanie noosfery*, op.cit.

⁴² Ibidem.

jektów programistycznych albo nie zostaje dokończonych, albo zostaje odrzuconych przez użytkowników⁴³. Oczywiście istotnym czynnikiem przyciągającym wolontariuszy do bezpłatnej pracy nad projektem jest charyzmatyczna osobowość „przywódcy”, który potrafi przyciągnąć do siebie rzesze entuzjastów. Przykładem takiej osoby jest L. Torvalds „energetyczny ekstrawertyk, czerpiący radość z tłumem ludzi, którego sposób mówienia i odruchy przypominają komika rozbawiającego publiczność”⁴⁴. Poświęcający swój wolny czas programiści, nie mogą czuć, że pracują pod presją czasu, w ogromnej korporacji, gdzie zniechęcony przez nich szef, czyha na ich potknięcia, by móc w każdej chwili ich zwolnić. Badania psychologów potwierdzają, iż jeśli przełożony szanuje autonomię swoich podwładnych, tj. przedstawia im różne możliwości decyzyjne, konsultuje z nimi swoje pomysły, to pracownicy mają wysoką motywację wewnętrzną. Kontrolujący przełożony, tj. taki, który używa wyłącznie poleceń, kar i nagród w sposób instrumentalny, mający wywołać określone reakcje, prowadzi do niskiej motywacji do wykonywanych zadań⁴⁵. Choć działający w ruchu *open source* rzadko kiedy mają jakiegoś formalnego przełożonego, jeśli nawet się on pojawia jako np. kierownik projektu, rzadko kiedy narzuca swoją wolę wolontariuszom, konsultując swoje działania z nimi. Inni programiści *open source* świadomie wybierają taką formę działania, gdyż kontestują biurokratyczną, hierarchiczną strukturę wielu firm informatycznych, w których liczy się jedynie racja przełożonego. To – *ethos ochotnika* – posługując się terminem Raymonda – pozwala im wykonywać swoje zadania z prawdziwą pasją, dla dobra całej społeczności. Istotnym czynnikiem jest przekonanie programistów *open source*, że mają wybór. Przekonanie o możliwości wyboru jest istotnym czynnikiem motywacyjnym, zwiększającym wytrwałość w wykonywaniu jakiegoś zadania⁴⁶.

Warto podkreślić, że w wielu przypadkach programiści *open source* mają wrogi stosunek do komercyjnego oprogramowania lub firm, które uważa się za dominujące na rynku⁴⁷. Ich postawa jest wyrazem entuzjizmu i antykomercyjnej postawy życiowej.

Tak naprawdę oprogramowanie pozbawione błędów jest jedynie pewnym typem idealnym, do którego należy dążyć, lecz jego osiągnięcie jest bardzo trudne. Z raportu *Software Lifecycle* wynika, iż w komer-

⁴³ E. Raymond, *Katedra i Bazar*, op.cit.

⁴⁴ Ibidem.

⁴⁵ W. Łukaszewski, D. Doliński, *Mechanizmy leżące u podstaw motywacji*, op.cit., s. 460–461.

⁴⁶ Ibidem, s. 452.

⁴⁷ E. Raymond, *Zagospodarowywanie noosfery*, op.cit.

cyjnym oprogramowaniu występuje od jednego do siedmiu błędów na 1000 linii kodu⁴⁸. Jednakże, jak wskazywał E. Raymond, przy dostatecznej liczbie patrzących, każdy błąd jest niegroźny⁴⁹. Raymond wskazuje, że programista, który potrafi korzystać jedynie ze swojego mózgu, pracując indywidualnie, będzie pozostawał w tyle za tym, który tworzy w sposób otwarty, licząc na ulepszanie kodu, dzięki zespołowej pracy⁵⁰. Dzięki zespołowej pracy wielu entuzjastów i wolontariuszy, szerokiej współpracy i swobodnemu dostępowi do informacji, kod Linuxa był lepszej jakości, niż wersje Unixa objęte prawem autorskim⁵¹. Programiści *open source* mogą przyjmować dwie orientacje motywacyjne na poziom wykonania lub na osiągnięcie mistrzostwa. Orientacja na poziom wykonania jest sposobem demonstrowania w określonej sytuacji kompetencji wyższych (lub przynajmniej nie niższych) od innych, zaś orientacja na osiągnięcie mistrzostwa kieruje jej zwolenników ku rozwojowi własnych umiejętności, kompetencji i osiągnięciu, dzięki temu jeszcze wyższego poziomu w przyszłości⁵². Obie te koncepcje mogą stawać się motywatorami działania uczestników projektów *open source*. Działalność ta bowiem niejednokrotnie bywa sposobem na pokazanie swoich wyższych umiejętności informatycznych od innych programistów tworzących wyłącznie komercyjne (dla wielu gorsze) oprogramowanie – orientacja na poziom wykonania. Programiści udział w projektach *open source*'owych traktują jako przepustkę do lepszej pracy, zdobycia uznania środowiska i wysokiej pozycji w nim, czasem sam udział gwarantuje ciekawe i opłacalne finansowo zlecenia. Najlepsi crakerzy⁵³, choć działający nielegalnie lub na granicy prawa, nierzadko otrzymują świetne posady w dużych korporacjach, gdyż wykazali się dużymi kompetencjami i umiejętnościami programistycznymi.

⁴⁸ <http://students.mimuw.edu.pl/SO/Projekt04-05/temat2-g4/wstep.html>.

⁴⁹ C. Meszyński, *Wpływ kultury hakerskiej na rozwój społeczeństwa informacyjnego. Hakerzy jako awangarda technologiczna*, op.cit., s. 154.

⁵⁰ E. Raymond, *Zagospodarowywanie noosfery*, op.cit., s. 44.

⁵¹ M. Castells, *Galaktyka Internetu*, op.cit., s. 57.

⁵² W. Łukaszewski, D. Doliński, *Mechanizmy leżące u podstaw motywacji*, op.cit., s. 464.

⁵³ Crackerzy (od *crack* – łamać) to osoby łamiące zabezpieczenia komputerowe.

Magdalena Szpunar

**The Internet and the Gift Culture.
The *Open Source* Movement Phenomenon**

The paper discusses the problems related to the *open source* programmers activity, which can be considered as contributing to the gift culture becoming the basic feature of new media. The author attempts to answer the question why thousands of people willingly, without any remuneration, devote their free time and family life opportunities in order to work for the sake of the open source idea. The author quotes arguments for the thesis that the acts undertaken by hackers are incompatible with the strongly materialistic and consumer-oriented contemporary culture, in which nearly all transactions are based on measurable material profits. The paper presents motivation mechanisms of volunteers who act within the *open source* movement, indicating simultaneously the benefits they gain from this activity.