OPEN SOURCE ЯК МОДЕЛЬ ДЕЦЕНТРАЛІЗОВАНОЇ КООПЕРАЦІЇ: ПРАКСЕОЛОГІЧНИЙ АНАЛІЗ ВІДКРИТОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

АЛІМ КАЛЮГА, студент, Фаховий коледж Класичного приватного університету (м. Запоріжжя, Україна)

ВОЛОДИМИР КОВТУН, викладач закладу вищої освіти кафедри інформаційних технологій та дизайну, Інститут управління Класичного приватного університету (м. Запоріжжя, Україна)

DOI: https://doi.org/10.36059/978-966-397-664-8-10

Вступ. Розробка програмного забезпечення традиційно асоціюється з чіткою організацією, розподілом ролей і централізованим управлінням проєктом. Проте феномен Open Source демонструє іншу модель: складні програмні продукти можуть створюватися й розвиватися завдяки добровільній взаємодії великої кількості незалежних учасників [3, с. 19–63; 7, с. 1–35]. Їх об'єднує не адміністративний примус, а власний інтерес, професійна мотивація та спільна мета [5, с. 197–234]. Саме тому відкрита розробка становить інтерес не лише як технологічне явище, а й як особлива форма децентралізованої кооперації [4, с. 369–446]. Праксеологічний підхід (найбільш ефективний) дозволяє розглянути Open Source через людську дію, індивідуальний вибір і добровільну взаємодію [2, с. 11–14] та з'ясувати, яким чином із множини самостійних рішень розробників формується здатна до самоорганізації система створення програмного забезпечення.

Мета та завдання дослідження. Метою дослідження є з'ясування механізму децентралізованої кооперації

у відкритій розробці програмного забезпечення з позицій праксеологічного підходу. Для її досягнення поставлено завдання визначити мотиви участі розробників в Open Source; дослідити, як індивідуальні рішення та добровільний внесок окремих учасників перетворюються на узгоджений колективний результат; з'ясувати роль відкритого коду, спільних правил і механізмів саморегуляції в координації розробки; оцінити Open Source як практичну модель взаємодії, здатної функціонувати без єдиного централізованого управління.

Результати дослідження та їх обговорення. Аналіз моделі Open Source показує, що відкритість програмного коду змінює не лише технічний спосіб створення програмного забезпечення, а й саму логіку взаємодії між його розробниками. Е. Реймонд, протиставляючи «соборну» і «базарну» моделі розробки, показав принципову відмінність між централізованим створенням програмного продукту та відкритою моделлю, заснованою на залученні широкого кола учасників [3, с. 19–63]. Учасники відкритого проєкту можуть самостійно визначати форму і масштаб своєї участі: створювати новий функціонал, виправляти помилки, перевіряти чужий код, удосконалювати документацію або використовувати наявне рішення як основу для власної розробки [6, с. 236–253; 7, с. 1–35]. Координація при цьому виникає не через обов'язкову участь кожного, а через поєднання індивідуальних рішень навколо спільного програмного продукту.

З праксеологічної позиції принципове значення має добровільність такої участі. У праксеології Л. фон Мізеса людська дія розглядається як цілеспрямована поведінка, у межах якої індивід обирає засоби для досягнення суб'єктивно значущої мети [2, с. 11–14]. Такий підхід дозволяє по-іншому пояснити й участь у Open Source. Розробник долучається до проєкту, оскільки пов'язує свою дію з певним очікуваним результатом:

вирішенням власної технічної проблеми, набуттям досвіду, професійною репутацією, удосконаленням продукту, яким користується, або реалізацією власної ідеї. Дослідження економіки Open Source також показують, що мотивація його учасників не зводиться до одного чинника і може поєднувати безпосередні та відкладені професійні стимули [5, с. 197–234]. Отже, колективний результат Open Source не заперечує індивідуальної природи людської дії. Навпаки, він виникає внаслідок координації великої кількості самостійних рішень, мотиви яких можуть істотно відрізнятися.

Водночас децентралізація не означає відсутності правил. Дослідження Open Source-проєктів свідчать про формування в них власних механізмів організації, участі та координації [6, с. 236–253; 7, с. 1–35]. Процедури внесення змін, перевірки коду, обговорення пропозицій, розподілу прав доступу та вирішення технічних суперечностей формують внутрішній порядок відкритої розробки. Тому коректніше говорити не про відсутність управління, а про розподілену координацію. Репозиторій, система контролю версій, механізми pull request і code review та правила конкретної спільноти створюють середовище, в якому незалежні дії розробників можуть бути перетворені на узгоджений програмний продукт.

Особливе значення має те, що знання в такій системі не зосереджене в одного учасника. Ф. Гаєк показав, що необхідне для прийняття рішень знання розподілене між окремими індивідами і не може в повному обсязі існувати в одному центрі. Ця закономірність виразно проявляється у відкритій розробці: один розробник краще розуміє архітектуру програми, інший виявляє вразливість, третій пропонує оптимізацію, а користувач може першим помітити помилку в реальних умовах експлуатації. Відкритість коду створює можливість залучати ці розосереджені знання до розвитку спільного продукту. Отже,

потенційна перевага децентралізованої розробки полягає не просто у кількості учасників, а в можливості використовувати знання, досвід та ініціативу, які неможливо повністю сконцентрувати в одному центрі прийняття рішень [1, с. 519–530].

З організаційної точки зору цей феномен наближається до того, що Й. Бенклер визначає як peer production — форму виробництва, засновану на децентралізованій індивідуальній діяльності, координованій без традиційної ринкової ціни або управлінської ієрархії [4, с. 369–446]. Open Source у цьому контексті є особливо наочним прикладом: програмний продукт може виникати як результат взаємодії учасників, які не утворюють традиційної фірми, не перебувають у єдиній вертикалі підпорядкування та можуть мати різні мотиви участі.

Разом із тим Open Source не слід ідеалізувати як систему, що автоматично забезпечує ефективну самоорганізацію. Дослідження FLOSS указують як на значні можливості такої моделі, так і на складність управління, координації, підтримки участі та забезпечення стійкості спільнот [6, с. 236–253; 7, с. 1–35]. Відкритий проєкт може залежати від порівняно невеликої кількості активних розробників, стикатися з конфліктами, нерівномірною якістю внесків, проблемами безпеки та дефіцитом ресурсів для підтримки коду. Це дозволяє сформулювати важливий результат дослідження: децентралізація не усуває потреби в координації — вона змінює механізм її здійснення. Місце прямого централізованого управління частково займають правила спільноти, професійна репутація, добровільний внесок, технічні процедури перевірки та розподілена відповідальність.

Таким чином, Open Source можна розглядати як практичну модель децентралізованої кооперації, в якій порядок не обов'язково задається наперед у вигляді детального

централізованого плану, а значною мірою формується у процесі взаємодії учасників [3, с. 19–63; 4, с. 369–446]. Праксеологічний аналіз дозволяє побачити за програмним кодом ширший механізм: індивідуальний вибір породжує добровільну участь, добровільна участь забезпечує обмін розосередженими знаннями, а їх координація за допомогою спільних технологічних та організаційних правил здатна створювати складний колективний результат. Саме ця властивість робить Open Source не лише способом розробки програмного забезпечення, а сучасним цифровим прикладом того, як індивідуальна дія може перетворюватися на децентралізовану кооперацію.

Висновки. Проведений аналіз дає підстави розглядати Open Source не лише як спосіб організації розробки програмного забезпечення, а як практичну модель децентралізованої кооперації. Її основою є самостійний вибір учасника: розробник визначає мету власної участі, обсяг внеску та спосіб взаємодії зі спільнотою, тоді як сукупність таких індивідуальних дій здатна формувати результат, масштаб якого перевищує можливості окремого учасника. Встановлено, що децентралізація Open Source не означає відсутності організації. Її особливість полягає в іншому механізмі координації, заснованому на відкритості коду, добровільності участі, розподіленому знанні, професійній репутації та спільно вироблених правилах взаємодії. Водночас ефективність такої моделі не є автоматичною: вона залежить від якості координації, активності спільноти та здатності учасників підтримувати життєздатність програмного продукту. Праксеологічний аналіз дозволяє виявити головну особливість Open Source: колективний технологічний результат виникає тут не всупереч індивідуальному вибору, а завдяки механізмам його координації. Відтак відкрита розробка програмного забезпечення може розглядатися як сучасний цифровий приклад перетворення добровільної індивідуальної

дії та розосередженого знання на складний колективний результат без необхідності повної централізації процесу його створення.

Список використаної літератури:

1. Hayek, F. A. The Use of Knowledge in Society. *The American Economic Review.* 1945. Vol. 35, No. 4. P. 519–530. URL: https://www.jstor.org/stable/1809376

2. Mises, L. von. Human Action: A Treatise on Economics. Scholar's ed. Auburn, Alabama: Ludwig von Mises Institute, 1998. 912 p. URL: https://mises.org/library/book/human-action

3. Raymond, E. S. The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary. Revised ed. Sebastopol, CA: O'Reilly Media, 2001. 256 p. ISBN: 978-0-596-00108-7. URL: https://www.oreilly.com/library/view/the-cathedral/0596001088/

4. Benkler, Y. Coase's Penguin, or, Linux and The Nature of the Firm. The Yale Law Journal. 2002. Vol. 112, No. 3. P. 369–446. DOI: 10.2307/1562247

5. Lerner, J., Tirole, J. Some Simple Economics of Open Source. *The Journal of Industrial Economics.* 2002. Vol. 50, No. 2. P. 197–234. DOI: 10.1111/1467-6451.00174

6. von Krogh, G., Spaeth, S. The Open Source Software Phenomenon: Characteristics That Promote Research. *The Journal of Strategic Information Systems.* 2007. Vol. 16, No. 3. P. 236–253. DOI: 10.1016/j.jsis.2007.06.001

7. Crowston, K., Wei, K., Howison, J., Wiggins, A. Free / Libre Open-Source Software Development: What We Know and What We Do Not Know. ACM Computing Surveys. 2012. Vol. 44, No. 2. Article 7. P. 1–35. DOI: 10.1145/2089125.2089127