

N. A. Sydorov

ECOSYSTEM RESEARCH IN SOFTWARE ENGINEERING

The application of the concept of the ecosystem in the software engineering shows the existence of the same problems regarding the definition of the concept of the ecosystem and its use for research that still exists in ecology. Justification for applying the concept of the ecosystem in an area that differs significantly from the ecology area, as in our case, requires researchers to look for analogies. This primarily applies to landscape, energy and matter transfer chains (trophic chains) and nutritional cycles. Until such analogies will be found in software engineering, the ecosystem research is nothing more than system analysis, and the concept of the ecosystem is an attractive concept. The purpose of this position article is to draw the attention of the software engineering community to ecosystem research. Three concepts of ecosystems in ecology, software and software engineering are considered. The composition and essence of ecosystem research in the software engineering are given. The literature review of on the state of ecosystem research in the software was carried out.

Keywords: Software engineering, software ecosystem, software engineering ecosystem, software engineering ecosystem cycles.

M.O. Сидоров

ЕКОСИСТЕМНІ ДОСЛІДЖЕННЯ В ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Застосування концепції екосистеми в інженерії програмного забезпечення показує існування тих самих проблем щодо визначення концепції екосистеми та її використання для досліджень, які все ще існують в екології. Обґрунтування застосування поняття екосистеми на території, яка суттєво відрізняється від території екології, як у нашому випадку, вимагає від дослідників пошуку аналогій. В першу чергу це стосується ландшафту, ланцюгів переносу енергії та речовини (трофічних ланцюгів) і циклів харчування. Поки такі аналогії не будуть знайдені в інженерії програмного забезпечення, дослідження екосистеми є не чим іншим, як системним аналізом, а концепція екосистеми є привабливою концепцією. Мета цієї позиційної статті — привернути увагу спільноти інженерії програмного забезпечення до дослідження екосистем. Розглянуто три концепції екосистем, а саме в екології, програмному забезпеченні та інженерії програмного забезпечення. Наведено склад та сутність екосистемних досліджень у інженерії програмного забезпечення. Проведено огляд літератури щодо стану екосистемних досліджень у інженерії програмного забезпечення.

Ключові слова: інженерія програмного забезпечення, екосистема програмного забезпечення, екосистема інженерії програмного забезпечення, екосистемні цикли інженерії програмного забезпечення

Introduction

The application of the concept of the ecosystem in the software engineering shows the existence of the same problems regarding the definition of the concept and its use for research that still exists in ecology [1-3]. Justification for applying the concept of an ecosystem in an area that differs significantly from ecology, as in our case, requires researchers to look for analogies. This primarily applies to landscape, the energy and matter transfer chains (trophic chains) and nutritional cycles. Until similar constructions are found in the software engineering, ecosystem research will

be nothing more than system analysis, and the ecosystem will be an attractive concept but without sense.

The article [4] showed the expediency of applying the concept of the ecosystem in the software while preserving essence of the biology ecosystem. In the article [5], the concept of the software engineering ecosystem introduced. The purpose of this article is to draw the attention of software engineering community to ecosystem research. To justify position, the types of ecosystems - the biology ecosystem, software ecosystem, software engineering eco-

system and features of the ecosystem research in software engineering are considered. The state of the ecosystem research in the software by the review of the relevant literature is presented.

To the using of the ecosystem concept

Consider three using of concept of ecosystem - the ecology ecosystem, the software ecosystem and the ecosystem of software engineering. Epistemology of the ecosystem concept can be found in article [1].

Ecology ecosystem. The term "ecosystem" should denote a group of organisms living in a certain area, interacting with their physical environment in such a way that the flow of energy leads to a well-defined trophic structure, biotic diversity and material cycles [6]. The concept of the ecosystem can be applied to various ecological objects. This application is justified by the fact that they demonstrate similar forms of relationships (energy and material cycles). The concept of the ecosystem is based on the concept of the system, but the ecosystem research differs from system research. This difference is expressed is that living components are necessarily present in the ecosystem and demands are made requirements that include study of internal cycles of matter, energy flows, and nutrient cycle, and due to the presence of non-living components, the ecosystem is characterized interdisciplinarity [6]. It is clear that the application of the concept in software engineering should be guided by the same requirements and a necessary condition for their satisfaction should be the presence of analogues of interactions specified for ecological ecosystems.

Software ecosystem. This a widespread term today that has many interpretations. In the review article [7], four definitions of software ecosystems are given. Analyzing these definitions, the authors identify three main elements that combine these definitions. This is platform, business and relationships. In addition, a generalized definition of the software ecosystem as the interaction of a set of subjects on top of a general technological platform, leading to a number of solutions in software or services is formulated. The article [8] has no new terms, but definitions are given that

are related to the software ecosystem: mobile learning ecosystems/mobile ecosystem, open source free software ecosystem, open ecosystem, digital (business) ecosystem. Here, these definitions stand next to the definition of the software ecosystem. As for cycles and chains, the work [7] indicates that the software ecosystem is an association that manages the ecosystem, executing the platform, creating and applying rules, processes, business procedures, setting and controlling quality standards and/or organizing relationships with the actor.

Software engineering ecosystems. In the work [5], based on the assumption that in software engineering, as well as in ecology, there should be a wider range of ecosystems than software ecosystems known in the literature, the concept of the software engineering ecosystems was introduced. The classification of ecosystems and the concepts of flows and chains similar to biological ecosystems was proposed. As in [5], in this article we will keep to the following point of view.

The ecosystem is a concept, not a real entity and is based on the concept of the system. The concept can be applied to biogeocenosis on the landscape that has system boundary, within the software engineering territory [5]. Each system, in the defined landscape of the software engineering territory that is open to material and energy flows and their analogies, and contains at least two living beings (organisms) and meet the requirements of the ecosystems, can studied as a software engineering ecosystem. However, since the researcher determines the boundary of the landscape based on the goals of the study that the previously determined boundary is not always the system boundary and biogeocenosis on the landscape is not will be a system. Further will be studied only system biogeocenosis. As an analogue of energy transferred in biological ecosystems in trophic cycles, it is proposed to use the concept of value based on the energy theory of value [9]. To determine ecosystems of the software engineering territory, we will use a structural ecosystem approach [10]. Therefore, in the ecosystem, in addition to the abiotic component and value, we consider four more types of elements of the ecosystem structure [10]. Namely, firstly, the types of activities that determine discrete actions that must be

carried out in order for the value proposal to materialize. Secondly, actors who are subjects and who carry out activities. Thirdly, positions that determine where subjects are in the flow of activity and characterize who transfers value to whom; fourthly, connections that determine transfers value and artifacts between subjects. There are activities at the center of the definition of the ecosystem. To make the ecosystem sustainable, actors need to interact in such a way that the value materializes. Therefore, as an analogue of the trophic chain, we will use the chain of interdependent value creation by actors performing activities in the landscape using and producing the abiotic components (artifacts). The activities of the actors play an important role in the ecosystem, defining its functions and services. Sometimes one actor can own the platform (a product, service or technology) and plays the leader role in the software engineering ecosystem then the ecosystem is called the platform ecosystem [11].

In the same way, we will consider software landscapes on which artificial biota can operate - computer programs, as well as landscapes on which both types of biota (human and computer programs) act [12]. For a system formed by artificial biota, a chain of formation of the emergence function of the system can be specified as an interdependent value creation chain.

The place the software engineering ecosystems in the living systems

The software engineering will be considered as part of human activity aimed at transforming the biosphere into an anthroposphere (noosphere). Transforming is carried out by human influence on the biosphere by his activity and thought. Although thought is not a form of energy, nevertheless under its influence, processes occur and artifacts are created that change the biosphere. Occupying a certain area of the biosphere, software engineering carries out a variety of activities aimed at solving practical tasks often occurring in other territories, thereby playing a special role in the context of the anthroposphere. In general, the following types of the software engineering activity on human activity habitats can be distinguished (Fig. 1):

- software engineering education and research. The habitat consists of universities and fundamental research organizations;
- software engineering. The habitat is organizations that transform the results of the fundamental research to recommendations for practical application;
- software development and maintenance. The habitat consists of organizations that create and maintain software products for different domains;
- software market. The habitat includes organizations supplying software products;
- software operation. The habitat is defined on other territories on that the software product is used;
- executing software. The habitat is the software products themselves.

It is obvious that these types of activities, although specific, are carried out in the context of the biosphere and, therefore, the biogeochemical cycles and the waste activities are inherent the territory of software engineering and they must be studied in the software engineering ecosystems (Fig. 1).

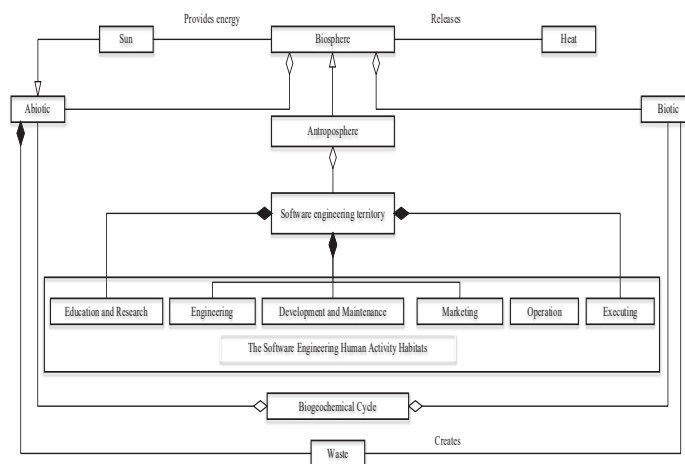


Fig. 1. The types of the software engineering activity on human activity habitats

Considering the territory of software engineering as part of the living system in which take place biogeocenosis, can apply to them the concept of the ecosystem to it and after that will study by performing the ecosystem research. Since the concept of a software engineering ecosystem is applied in the context of biological ecosystems that the relevant parts of both biotic and abiotic components, as well as biogeochemical cycles taking place

in the territory of software engineering, may be of interest also. However, new aspects of software engineering ecosystem research should be related with locating of the territory in the anthroposphere. Here the anthropos are new biotic components and the anthropogenic artifacts are new abiotic components, and arise the new networks and cycles inherent only in the software engineering ecosystems (Fig. 2). Therefore, along with the known ones (from ecology), there should be new directions of the software engineering ecosystem research.

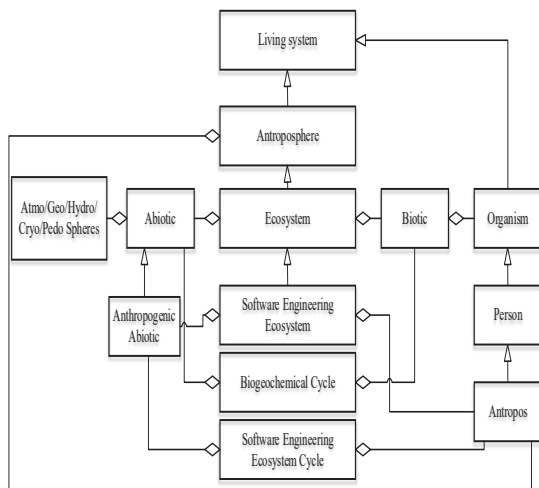


Fig. 2. Software engineering ecosystems in the anthroposphere

Thus, in living systems of anthroposphere, the object of the ecosystem research in the territory of the software engineering will be the human activity habitats and its social systems (Fig. 1, 3) [5]. Applying the concept of the ecosystem, these systems can be study as the software engineering ecosystems, taking into account the additional components of biotic is anthropos and abiotic are antropogenic artifacts that are inherent in the anthroposphere. In the software engineering ecosystem, except for artifacts that consist of assets and tools [13], data, information and knowledge should be considered, as well as the software product as territory for the software ecosystem (Fig. 3).

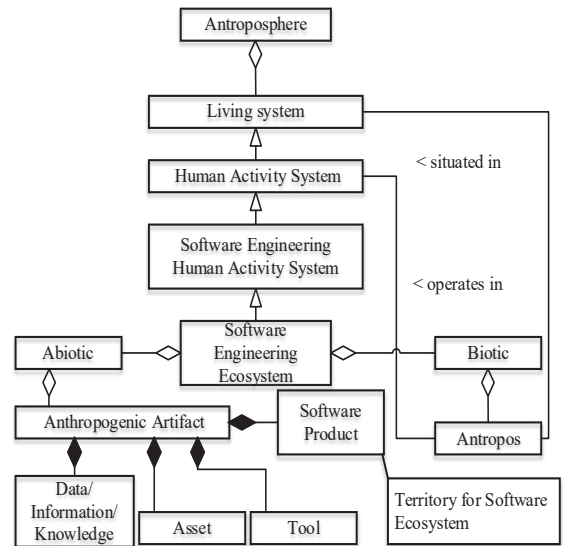


Fig. 3. Software engineering ecosystem

The ontology of the software engineering ecosystem with its attributes is shown on Fig. 4.

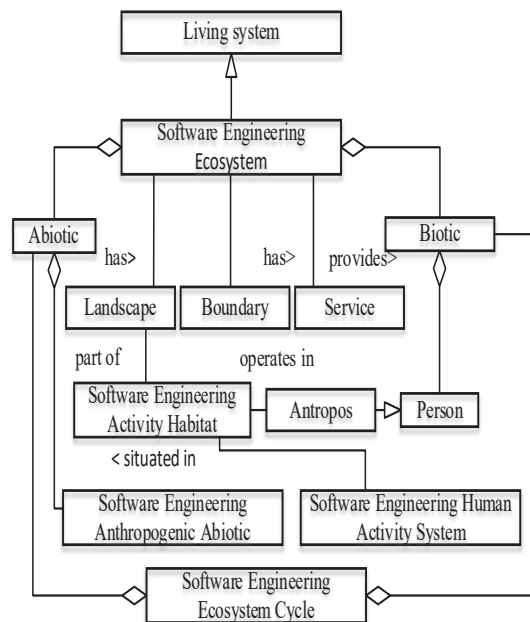


Fig. 4. The ontology of the software engineering ecosystem

In the context of the software engineering habitats the activity of anthropos to lead to implementing processes that can be a linear (chains), nonlinear (networks) and repetitive (cycles). These processes must be objects for the ecosystems study regardless of the software engineering ecosystem type [5]. They are the following chains, cycles and networks (Fig. 5):

- software engineering life cycle, this is the main cycle in software engineering,

therefor inherent in application development human activity.

- cycles of data, information and knowledge, especially inherent in such human activity as software engineering education and research, and software research.

- software reuse cycle, analogous to nutritional cycles, here, these are cycles of reuse of software life cycle artifacts (requirements, design, code, and documentation) and knowledge artifacts.

- value chain/web, this is a value chain and network - an analogue of energy chain in the software engineering ecosystem [9].

- waste cycle is a waste creation and disposal cycle using 3R (Reuse, Rework, and Recovery).

All of the above relates to the application of the ecosystem concept in the context of living systems [14]. In the paper [5], it is

shown that the software ecosystem is the application of the concept of the ecosystem to the software product is the habitat of in the sense of artificial life using analogues of the biotic component of biological ecosystems [4, 15, 16] (Table 1). Here, there are more questions than answers. Therefore, next only software engineering ecosystem research in the context of human activity habitat (living system) will be considered further.

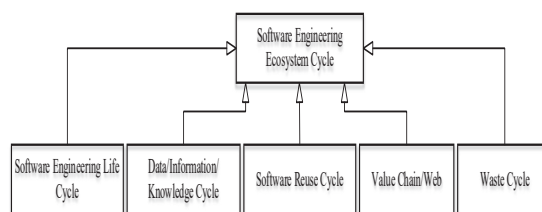


Fig. 5. Software engineering ecosystem chains, cycles and networks

Table 1. Analogues of biotic components of Software ecosystem for biotic components of Biology ecosyste

№	Software ecosystem	Biology ecosystem
1.	Characters, lexemes, simple types	Molecules (?)
2.	Operators, complex types, Structured operators	Tissues
3.	Subroutines (algorithms)	Cells (?)
4.	Modules (Classes), mega modules (Subsystems)	Organs (organisms)
5.	Programs	Genus
6.	Computer programs	Species
7.	Software	Subspecies
8.	Software of domain	Varieties
9.	Software product	Individual of the Varieties
10.	Software products of same type on the domain territory	Population
11.	Software product line (family)	Evolving Population
12.	Populations of the Software products on the some domain territory	Community (Biocenosis)
13.	Populations of the Software products on the some domain territory and abiotic	Biogeocenosis

14.	Software products as systems	Ecosystems
15.	Information flows	Energy flows (?)
16.	Control (algorithms)	DNA, RNK (?)

Ecosystem research

In accordance with the goals of studying objects in software engineering using the concept of the ecosystem, we will use traditional approaches to studying systems [3]. However, need take into account the following features:

- Structural analysis is performed on biotic and abiotic components. In addition, structural analysis will have study the biotic component as a community, including its various forms, such as the working community, the program community, and abiotic component as, for example, the artifacts populations.

- Functional analysis is aimed at studying the added value chains instead of food chains, and circulation chains of artifacts of the legacy software instead of nutritional cycles. For software engineering ecosystems with a human community in the biotic component, and possibly software ecosystems, flows that lead to the emergence and elimination of waste should be investigate.

- System analysis, as in the biological ecosystem, aims to study the emergence, stability, self-organization, evolution and sustainability of the ecosystems.

Ecosystem research in the software engineering

Next, based on research on biological ecosystems, we will consider the main areas of research applying the concept of the ecosystem in the software engineering. The research will be divided into two groups - basic and target (Fig. 6). Basic studies are those aimed at collecting data and information about the ecosystems and are divided on long-term study and the ecosystems history study. Targeted research includes structural, functional and system analysis.

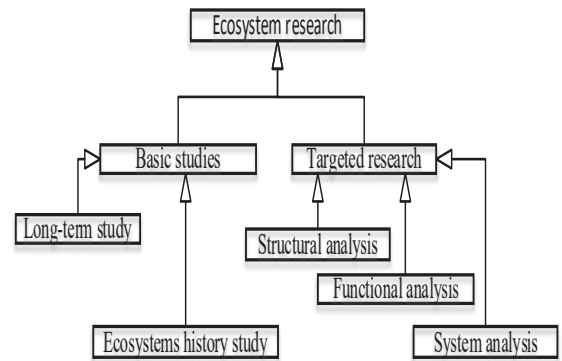


Fig. 6. Software engineering ecosystem research

Basic studies are those aimed at collecting data and information about the ecosystems. This will provide more complete and reliable results of targeted studies. Long-term studies and historical research are the basic research.

Long-term study is research conducted in the ecosystem, continuously for long periods of time [17, 18]. The duration of research should be at least such as long as the dominant event of the ecosystem occurs or until the event under study changes [18]. In article [19], a classification of types of long-term ecological studies is provided. There are retrospective research (passive - deep and recent; active - opportunistic, intentional and remodeled) and prospective research. Research data can be directed to the following: to develop and test the theory; to provide general knowledge about the operation of selected ecosystems; to provide student and postgraduate training [17]. The long-term ecological studies have features that consider before starting research. It is, firstly, permanent providing of money, time, staff, means; secondly, keeping research from complacency, if funding job is provided in the long run; third, limit by practical considerations, for example, by time [17]. Apparently, these features need

take into account for the software engineering ecosystem research. The long-term studies expedient when there are the slow processes, rare events, complex phenomena [17]. The field study and the field experiment can be used for the long-term ecological studies.

The ecosystems history study. As the structure, functions, services, landscape, and terrain of the ecosystems change over time, the ecosystems have a history. For quantitative and especially comparative studies, it is critical to determine the current state of the ecosystem and state changes over time. For example, how the transformation of the ecotope into a biotope of the software engineering ecosystem was carried out. For historical research, the empirical or natural-historical method, the field study, the anthropological and ethnographic studies can be use.

Targeted research includes structural, functional and system analysis.

Structural analysis aims to study the following components of the ecosystem as structure [10]. Firstly, activities, which specify the actions for the value proposition to materialize. Secondly, creating by the actors. Thirdly, positions that determine places where actors are in the flow of activity. Fourthly, links that determine the transfer of value between the actors. The essence of the value may vary - material, information, influence, funds. Need to be considered that there is the transfer of energy in the ecosystem, biotic and abiotic are characterized by diversity and therefore need multidiscipline study, and that the structure of the ecosystem can be multi-level. The field study, the monitoring and classification methods can be used for the structural analysis. The software engineering ecologist must follow an example the ecologist in studying species of both biotic and abiotic component of the software engineering ecosystem [20].

Functional analysis contributes to the identification of functions, services and interactions (streams and chains) aimed at creating value.

Ecosystem functions on the definition of ecosystem processes are based, which can be interactions inside biota and between the biotic and abiotic components of the ecosystem, covering material cycles and energy flows. The functions of the ecosystem can be defined as

the ability of processes and components of the structure to provide goods and services that meet human needs, directly or indirectly. Article [21] examines the historical development of the concept of ecosystem services in economic theory and practice, which can be useful in the software engineering.

Ecosystem services are defined as a set of ecosystem functions that are useful for humans [22]. Thus, ecosystem services in software engineering should include the results of performing ecosystem functions that are beneficial to humans. In the software engineering, as with other types of the ecosystems, these functions can be divided into three groups - economic, sociocultural and environmental [22]. The first group, depending on the type of ecosystem, includes knowledge, skills, products and artifacts. The second group is determined by the presence of people in the biota of the ecosystem that bring socio-cultural values to the concept of the ecosystem. Both are related to not only the culture of software engineering, but also to society and culture in a broad sense. The third group, these are functions regarding the ability of ecosystems to regulate flows that lead to waste disposal [4]. An important aspect in the study of ecosystem services is their evaluation. The papers [23, 24] offer models for evaluating ecosystem services that can be applied to software engineering ecosystems.

Researching flows, chains and cycles that correspond to ecosystem functions is an important part of the functional ecosystem study. For the software engineering ecosystems, these are the study of the added value chains (food chain analogues), reuse cycles of the legacy software artifacts, knowledge, practices (analogues of nutritional cycles). In biological ecosystems, the nutrient cycle is one of the most important processes, as it ensures the transformation of "old" (legacy) matter into nutrients used by living organisms. It also maintains the balance of those nutrients that are necessary to maintain a healthy ecosystem. Biological ecosystems consider four cycles of nutrients (water, oxygen, nitrogen, carbon). In software engineering ecosystems, as an analogue of such cycles, we suggest using the software engineering artifacts (knowledge, policies, software artifacts, best practices) reuse

cycles (Fig. 4). Without reuse cycles, software engineering ecosystems can become unbalanced and unhealthy, similar to biological ecosystems, leading to inefficiency and disintegration. In addition to them, flows that lead to generated and disposal of waste should be studied. Waste cycles are important and arise for various reasons, for example, waste is a result of inappropriate organization of the activity process. For functional analysis, the technique of system functional analysis can be used. For value chains and cycles, it may will be useful to apply value chain analysis [25, 26]. System analysis is aimed at studying a part of the world as a system localized within the ecosystem. Of particular interest are the emergent functions of the system and modeling and simulation. Since ecosystems are usually open systems, connections between the ecosystem and other parts of the world presented as ecosystems are also studied. For system analysis, experimental simulation, simulation experiment, conceptual modeling, simulation should be used. Finally, it should be noted that the ecosystem research differs from systems research in that it is interdisciplinary.

Literature Review

The literature review was performed. The goal of review is find out the state of the ecosystem research in context of the software. For this goal are formulated the following two research questions:

(RQ1) Do software ecosystem researchers use biological analogies, and if so, which ones?

(RQ2) What type of the ecosystem research have software ecosystems researchers shown interest in?

Below are the results of the analysis.

RQ1. Do software ecosystem researchers use biological analogies, and if so, which ones?

In the works [16, 27] are attempts to use analogies. For example, [27] offers two types of biota and an analogue of the food chain for the software ecosystem. The first type is the participants of the software project, which are represented by four categories - core developers, active developers, peripheral developers and users. Between them, a connection is indicated, according to the authors, sim-

ilar to a trophic chain directed from core developers to users. The second type are the software components and the project contributors that are compared whit the biological species in the biological ecosystem. Indicated that as in the trophic network, some of these components (such as joint development platforms and software libraries) act as producents. A wider range of analogues of components for this type can be found in the work [16]. However, it has no analogies with chains and cycles.

RQ2. What type of the ecosystem research have software ecosystems researchers shown interest in?

Basic studies (Long-term studies). The paper [28, 29] is work from this type of research. Author studied three aspects of the software ecosystems - how this ecosystem has emerged; how the present organization works in terms of its structure, its processes, and its product line, and how the organization relates to actors in its external environment.

Basic studies (History of ecosystems). The paper [29] presents the results of studying how the object-oriented classes are used in software packages over time to increase understanding of the evolution of the software ecosystem, as well as the maintenance and survivability of projects.

Targeted research (Structural analysis). The article [30] considers a subset of communities found on GitHub and a variety of roles is identified. The article [31], bots as parts of biotic are used to coordinate open source software projects. Four different classes of the bots that coordinate work in such projects (broker, checker, gatekeeper, and manager) have been identified. In the articles [32, 33], the workforce ecosystem as a structure focused on creating value for the organization was defined.

Targeted research (Functional analysis).

- Ecosystem functions and services. In the article [34], proposed an open source online platform (RECOSystem) to support the development and management of recommender systems. In the article [35], the definition of the software testing ecosystem was proposed. The article [36], described the component-based software development ecosystem and its main service is the supply of compo-

nents to the market. The article [37], described the ecosystem of scientific software, as well as the production of scientific software components. In the article [38], the program analysis platform that aims to build a data-driven ecosystem is presented. The article [39], presents an ecosystem that Ericsson has developed for the systematic practice of large-scale reuse of microservices in a cloud context. The article [40] draws attention to decision-making on the design of software and systems for products, processes, projects that maximize the value created for invested resources. In the article [41], the concept of a unified software value chain and the first empirical proof of the concept are proposed. This article examines the value chain in the context of the software life cycle. This view supports our view of software engineering ecosystems. In the article [42], the concept of the software ecosystem service is discussed.

- The reuse cycles of the legacy software. In the article [43], the concept of the ecosystem applies to the software industry to encourage developers to reuse and multiple use the software components. This view cannot be directly related to the cycles of reuse of the legacy software, but it can be accepted to study these cycles. In the article [44], value networks for DevSecOps are presented.

- Flows of waste generation and disposal. The work [45] is the most famous in the software engineering about this theme. This aspect of the ecosystem research can certainly be related to issues of the sustainable and environmental software engineering [46].

Targeted research (System analysis).

The article [47] the structures of known ecosystems, as well as formulated and analyzes problems related to openness, management, analysis and quality in such ecosystems are presented. The article [48] draws attention to orchestration in software ecosystems and offers a new understanding of it. The article [49] provides a systematic display of the description and documentation of the software ecosystem through modeling.

Conclusion

This work is a continuation of the author's works [4, 5, 50 - 52]. Represents the author's view on the problem of applying the con-

cept of the biology ecosystem in software engineering. Attention is drawn to the inadequacy of the existing application of the concept to the software ecosystem compared to the biological approach. That can be explain by the fact that the software ecosystem uses an approach that considers ecosystems as networks, organized around a keystone species and characterized by a large number of loosely interconnected participants who depend on each other for their mutual effectiveness and survival. In paper [10], such approach was called "ecosystem-as-affiliation", and in paper [11], it was called "platform ecosystems". At the same time, the boundaries of the landscape and itself landscape are not specified and there are no analogues of chains and cycles, the presence of which determines the essence of biological ecosystems. Therefore, now the software ecosystem, in our opinion, is nothing more than a fashionable term [53]. In this article, attention is drawn to a different approach "ecosystem-as-structure", when ecosystem is configuration of activities that are defined by a value proposition. In this context, attention is drawn to the ecosystem research and the need to find analogies that are known in ecology.

References

1. R. V. O'neill, Is it time to bury the ecosystem concept (with full military honors, of course!), *Ecology*, 82(12), (2001) 3275–3284.
2. E. Becker, B. Breckling, Border zone between ecology and systems theory, *Ecology Revisited*, Springer, 2011, 385–403.
3. A. Mittwollen, Unity in Ecology? An Investigation of Patterns, Problems and Unifying Concepts of Population Ecology, Systems Ecology and Evolutionary Ecology (Biology), Ph.D. thesis, Bremen, 2002.
4. N.A. Sydorov, Software ecology, *Software engineering*, 1 (2010), 53-61 (in Ukrainian).
5. N. Sydorov, Software Engineering Ecosystems, in: *Proceedings of the 13th International Scientific and Practical Programming Conference UkrProg 2022Kyiv, Ukraine, October 11-12, 2022*.CEUR-WS: 10-Oct-2023, Vol-3501, pp. 91-100.
6. E P. Odum, *Fundamentals of Ecology*, Saunders Company, Philadelphia-London, 1971.
7. K. Manikas, Software ecosystems – A systematic literature review, *The Journal of Systems and Software* 86 (2013), 1294–1306.

8. O. Barbosa, C. Alves, A Systematic Mapping Study on Software Ecosystems, in: *Proceedings of the Workshop on Software Ecosystems*, 2011, pp. 15-26.
9. R. Costanza, C. J. Cleveland, Value theory and energy, in: *Encyclopedia of energy*, 6, 2004, pp. 337 - 346.
10. R. Adner, Ecosystem as Structure: An Actionable Construct for Strategy, *Journal of Management* Vol. 43 No. 1, January (2017), 39–58.
11. K. Kapoor, A. Ziaee Bigdeli, Y.K. Dwivedi, A socio-technical view of platform ecosystems: Systematic review and research agenda, *Journal of Business Research* 128 (2021) 94–108.
12. D. H. Ackley, Real artificial life: Where we may be, in: *The Proceedings of Artificial Life VII*, Portland, Oregon, August, 2000.
13. E. Zabardast, J. Frattini, Assets in Software Engineering: What are they after all?, *Journal of Systems and Software* 193(2022), 111485.
14. F. Capra, P. L. Luisi, *The systems view of life: a unifying vision* / University Printing House, Cambridge, United Kingdom, 2014.
15. D. H. Ackley, Real artificial life: Where we may be, in: *The Proceedings of Artificial Life VII*, Portland, Oregon, August, 2000.
16. T. N. Nguyen, *The Ecology of Software: A Framework for the Investigation of Business-IT Integration Issues and Trends of Information Technology Management in Contemporary Organizations*, in: *Proceedings of the Information Resources Management Association International Conference*, 2002.
17. D. Strayer, J. S. Glitzenstein, Long-Term Ecological Studies: an Illustrated Account of Their Design, Operation, and Importance to Ecology, in: *Occasional Publication of The Institute of Ecosystem Studies* Number 2, August, 1986.
18. G. E. Likens, *The ecosystem approach: its use and abuse*, Germany, 1992.
19. S. T. Jackson, Repurposing long-term ecological studies for climate change <https://doi.org/10.1073/pnas.2314444120>.
20. D. Kirsh, Explaining Artifact Evolution. in: *Cognitive Life of Things: Recasting the Boundaries of the Mind*. Malafouris, L. (Ed.), McDonald Institute for Archaeological Research, Cambridge, April 7-9, 2009. Cambridge University press.
21. E. Gómez-Baggethun, The history of ecosystem services in economic theory and practice: From early notions to markets and payment schemes, *Ecological Economics*, vol. 69 (2010), 1209-1218.
22. R. S. de Groot, A Typology for the Classification Description and Valuation of Ecosystem Functions, Goods and Services, *Ecological Economics*, *Ecological Economics* Volume 41, Issue 3, (2002) 393-408.
23. L. Hein, K. van Koppen, Spatial Scales, Stakeholders and the Valuation of Ecosystem Services, *Ecological Economics*, 57, (2006), 209–228.
24. J. J. Kaufman, *Stimulating innovation in products and services: with function analysis and mapping*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2006.
25. J. R. Wixson, Function Analysis and Decomposition Using Function Analysis Systems Technique, in: *International Council on Systems Engineering Annual Conference (INCOSE '99)*, 1999.
26. B. Boehm, *A Value-Chain Analysis of Software Productivity Components*, TRW Inc., 1976.
27. T. Mens, P. Grosjean, The Ecology of Software Ecosystems, *Computer*, October (2015), 112-114.
28. G. K. Hanssen, A longitudinal case study of an emerging software ecosystem: Implications for practice and theory, *The Journal of Systems and Software* 85 (2012) 1455–1466.
29. T. Mens, M. Claes, P. Grosjean, *ECOS: Ecological Studies of Open Source Software Ecosystems* (2014), Antwerp, Belgium 978-1-4799-3752-3/14, 2014.
30. D. Mendez, B. Baudry, M. Monperrus, Empirical Evidence of Large-Scale Diversity in API Usage of Object-Oriented Software, in: *Proceedings of IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, July 2013.
31. F. Hukal, N. Berente, Bots Coordinating Work in Open Source Software Projects, *Computer*, 52(9) (2019), 52-60.
32. E.J. Altman, J. Schwartz, D. Kiron, R. Jones, and D. Kearns-Manolatos, *Workforce Ecosystems: A New Strategic Approach to the Future of Work*, MIT Sloan Management Review and Deloitte, April 2021.
33. A. T. Gorski, I. Gligorea, Workforce and workplace ecosystem – challenges and opportunities in the age of digital transformation in: *Proceedings of the International Conference knowledge-based organization*, Vol. XXVIII ,No 1, 2022.
34. A. Abdalla, A Software Ecosystem platform for the development of Recommender Systems, June, 2020, DOI: 10.21203/rs.3.rs-34335/v1.

35. I. Santos, Software testing ecosystems insights and research opportunities, in: Proceedings of the XXXIV Brazilian Symposium on Software Engineering, 2020, pp. 421–426.
36. J. Arndt, J. Dibbern, The Tension between Integration and Fragmentation in a Component Based Software Development Ecosystem, in: Proceedings of the 39th Hawaii International Conference on System Sciences, 2006.
37. J. Howison, Understanding the scientific software ecosystem and its impact: Current and future measures, Research Evaluation 24 (2015), 454–470.
38. C. Sadowski, Tricorder: Building a Program Analysis Ecosystem, in: Proceedings of the 37th IEEE International Conference on Software Engineering, Florence, Italy, May 2015, pp. 598-608, DOI: 10.1109/ICSE.2015.76.
39. M. Usman, D. Badampudi, An Ecosystem for the Large-Scale Reuse of Microservices in a Cloud-Native Context, IEEE Software, (39)5, Sept.-Oct., 2022.
40. B. Boehm, Software Economics: A Roadmap, March 2000, doi: 10.1145/336512.336584.
41. A. Pussep, The Software Value Chain as an Analytical Framework for the Software Industry and Its Exemplary Application for Vertical Integration Measurement, in: Proceedings of the Seventeenth Americas Conference on Information Systems, Detroit, Michigan August 4 -7, 2011.
42. K. Manikas, K. M. Hansen, Software ecosystems – A systematic literature review, the Journal of Systems and Software, 86, (2013), 1294–1306.
43. J. Arndt, J. Dibbern, The Tension between Integration and Fragmentation in a Component Based Software Development Ecosystem, in: Proceedings of the 39th Hawaii International Conference on System Sciences, 2006.
44. J. Morales, R. Turner, Guide to Implementing DevSecOps for a System of Systems in Highly Regulated Environments, Technical Report CMU/SEI-2020-TR-002, 2020.
45. M. Poppendieck, T. Poppendieck, Implementing Lean Software Development: An Agile Toolkit, Addison-Wesley, 2003, 240p.
46. E. Kern, Green software and green software engineering - definitions, measurements, and quality aspects, in: Proceedings of the First International Conference on Information and Communication Technologies for Sustainability, ETH Zurich, Feb, 14-16, 2013.
47. J.V. Joshua, Software Ecosystem: Features, Benefits and Challenges, International Journal of Advanced Computer Science and Applications (IJACSA), 8(4), (2013), 242-247.
48. L. M. Maruping, S. Matook, The evolution of software development orchestration: current state and an agenda for future research, European Journal of Information Systems, 29:5 (2020), 443-445, DOI: 10.1080/0960085X.2020.1831834.
49. O. Pettersson, J. Andersson, A Survey of Modeling Approaches for Software Ecosystems, Springer International Publishing Switzerland, 2016, pp. 79-93.
50. N. Sydorov, Programming Style as an Artefact of a Software Artefacts Ecosystem, in: Proceedings of the Advances in Computer Science for Engineering and Education Applications, Springer, Cham, 2021, 244-255.
51. N.A. Sydorov, N. N. Sydorova, E.N. Sydorov, Description model of programming style ecosystem, Problems in programming, 2-3(2022), 74-81.
52. N.A. Sydorov, Toward a software artifacts ecosystem, Problems in programming, 4 (2020), 11.
53. S. Hyrynsalmi, S. M. Hyrynsalmi, Ecosystem: A Zombie Category? 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC).

Одержано: 25.10.2024

Внутрішня рецензія отримана: 01.11.2024

Зовнішня рецензія отриман: 02.11.2024

Про автора:

Сидоров Микола Олександрович,

д.т.н., професор

ORCID 0000-0003-0800-1668

Місце роботи автора:

НТУ України «Київський політехнічний інститут імені Ігоря Сікорського»,

кафедра інформатики

та програмної інженерії,

Моб. тел.: 067 7980361.

E-mail: nyksydorov@gmail/com