Федеральное государственное автономное образовательное

учреждение высшего образования

Национальный исследовательский университет

«Высшая школа экономики»

Высшая школа бизнеса

Дробышевская Екатерина Сергеевна

**Процесс создания и управления знаниями в Open-Source проектах с точки зрения теории самоорганизованной критичности.**

Бакалаврская работа

по направлению подготовки 38.03.05/2 Бизнес-информатика/Менеджмент

образовательная программа «Бизнес информатика»

Научный руководитель

д-р физ. мат. наук,

профессор кафедры управления информационными системами и цифровой инфраструктурой НИУ ВШЭ

Дмитриев А.В.

Москва 2022

[Introduction 3](#_Toc103714136)

[Chapter 1. Literature Review 6](#_Toc103714137)

[Knowledge 6](#_Toc103714138)

[Historical background 6](#_Toc103714139)

[Modern concepts and framework 8](#_Toc103714140)

[OS project community 11](#_Toc103714141)

[Community as a major OS project success factor 11](#_Toc103714142)

[Open-Source Economics 14](#_Toc103714143)

[Statistical properties of OS community 15](#_Toc103714144)

[The SOC concept and defining features 19](#_Toc103714145)

[Definition and history 19](#_Toc103714146)

[SOC models 23](#_Toc103714147)

[Related frameworks 24](#_Toc103714148)

[Self-Organization without Criticality 24](#_Toc103714149)

[Brownian Motion and Classical Diffusion 25](#_Toc103714150)

[Percolation 25](#_Toc103714151)

[Chaotic Systems 26](#_Toc103714152)

[Forced Self-Organized Criticality (FSOC) 26](#_Toc103714153)

[Network Systems 26](#_Toc103714154)

[SOC and knowledge flows 27](#_Toc103714155)

[Applications 29](#_Toc103714156)

[Chapter 2. Applications 31](#_Toc103714157)

[Theoretical framework 31](#_Toc103714158)

[Platform choice 31](#_Toc103714159)

[Knowledge development process 33](#_Toc103714160)

[Model validation 38](#_Toc103714161)

[Process semantics 38](#_Toc103714162)

[Experimental evidence 40](#_Toc103714163)

[Methodology 40](#_Toc103714164)

[Results and discussion 41](#_Toc103714165)

[Problem Level 41](#_Toc103714166)

[Repository Level 45](#_Toc103714167)

[**Conclusion** 47](#_Toc103714168)

[**Resources** 48](#_Toc103714169)

# Introduction

Born in n the end of 1980s OS movement proved its viability through a series of successful projects including the most widespread operation software platform and DMSs giants. Variety of business models has been built on OS approach marking wide appreciation among non-commercial and commercial organizations. Recently, inner-source practice evolves as more conservative in terms security modification that still retains some OS benefits, such as code base reuse, cross-team contribution, transparency and etc. More and more companies nowadays adopt either of the approaches the two or combination of any kind. On the consumer side then, an article provided by BCG indicates that 80% of respondents plan to increase their usage of open-source software next year, and 95% of IT specialists consider open-source to be of strategically importance[[1]](#footnote-2). As far as Russia is concerned, according to survey conducted by Accenture and Skolkovo foundation more that 90% of Russian companies are forecast to use OSS by 2026. The OSS adoption rate occurred to be fairly high rising to over 80% for IT-infrastructure and 60% for IT-support respectively. More than a half of companies declared that they participate in adjusting the OSS they use or contributing to the original projects. Although only 15% of the respondents were OS project maintainers themselves. Indeed, developing an OS strategy present a major challenge, no matter whether a company chooses to be project maintainer, contributor, or a regular user. Regardless of the position taken it still needs to manage OS software’s use. One popular practice is conducting workshops for employees that cover most prominent aspects. An alternative one is establishing open communities of excellence in order to foster collaboration and practice sharing. [[2]](#footnote-3)

On the state level, Russian Federation officials consider OS to be a strategic assets that allows for developing digital sovereignty. In 2010th OSS transition act addressing federal executive body and federal budgeted agencies was released[[3]](#footnote-4). Then, the government decree calling for software import substitution followed in 2015[[4]](#footnote-5). In 2021 digital autonomy was declared a primary focus in national OSS development program[[5]](#footnote-6). The paper states that fostering human capital, facilitating best practices spread, stakeholder engagement and providing targeted resource support for OS projects should be made a priority especially when it comes to communities that are not limited to developer professionals. I recon human capital development and best practice transfer are mutually interdependent as they deal with information and knowledge diffusion withing project communities. During open session on the program presentation, much emphasis was given to educational and raising awareness aspects. Consequently, national OSS foundation was established, and first national OS projects hackathon curated by leading Russian IT-companies and federal bodies all together for high school and university students was announced to be held in spring 2022.

In conclusion, development of OS culture is now assumed to be of strategical importance equally on corporate and state level. Knowledge exchange in its turn is recognized to be a vital part of the venture success. Likewise, the aspect was emphasized during OS national strategy project presentation, as well as the notion was included in text of final document[[6]](#footnote-7). Meanwhile, OS remains unexplained in terms of collaborative dynamics, as a result the credibility present of existent monitoring technics is doubted (Sunindyo et al 2013). A number of papers has already been devoted to presenting knowledge and innovation development from the SOC perspective that offered quite well-fitted model (Tadic et al. 2017, Lambiottea & Panzarasab 2009), though, practical guidance is lacking.

Therefore, the object of the study is OS community dynamics, and the subject is particularly knowledge developing. The purpose then is to develop an interpretable model of knowledge development dynamics in OS communities and examine possible mathematically derived characteristics in order to obtain sound performance metrics. With the goal in view following objectives were set:

1. Conduct a comprehensive review of knowledge management, OS studies and SOC-theory domains.
2. Develop a generic approach that reflects peculiarities of knowledge development process in OS community.
3. Collect empirical data.
4. Perform explorative analyses of the collected data in order to test whether associated statistics is consistence with SOC.
5. Provide interpretation of the obtained results

The paper is structured as follows. The first chapter includes literature review and the remarks that establish connection between three field of study. To the beginning, I provide a brief overview on the knowledge related research in regard to business activities and social value, so as to introduce framing approach. Next, the framework is applied to OS community dynamics addressing both semantic and statistical aspects of knowledge creation process. Finally, I explain the way statistical properties associated with OS project community dynamics are consistent with SOC phenomena. The provide a unifying approach drawn on the literature review that will serve as a basis for quantitative survey conducted on empirical data. Obtained results and accompanying discussion concludes the paper.

# Chapter 1. Literature Review

## Knowledge

### Historical background

Knowledge is «information that enables action and decisions, or information with a direction» (Becerra-Fernandez 2008, chapter ). Knowledge sharing has been targeted at collective perfection of knowledge by continuous interpretation, life-long learning underpinned via fusion of mutual teaching and learning and individual contemplation (McGrath, 2005). Indeed, knowledge development is a synergistic process. (Zeldin 1999). Ever since prehistorical times it was intended to benefit the wider community through best practice spread and adoption regardless of the ability of individuals comprehend such a concept. [Tojo Thatchenkery](https://www.semanticscholar.org/author/Tojo-Thatchenkery/2659017), [D. Chowdhry](https://www.semanticscholar.org/author/D.-Chowdhry/100709460) provide the knowledge timeline diagram reflecting major phases the researchers identified based on knowledge transfer means, scope of social systems, competency distribution, social strata the process was driven by, and level of knowledge sharing appreciation. (Thatchenkery & Chowdhry 2007).

Schematic

Description automatically generated with medium confidence

Figure 1 - Knowledge sharing timeline (Thatchenkery & Chowdhry 2007)

Tracing the story back to hunters & gathers period is not as farfetched as it may seem given that OSS key advantages rely upon voluntary contribution made for the sake of one’s own benefit. Thus, the way direction of value co-creation within developers and users’ community has much more in common with society of the time that those powered by robust governance. Namely, it is information exchange that drivers development of a collective memory (myths in a broader sense) substituting organizational vision, therefore keeping all members on the same course without vital restraining to individual initiative. Resulting tribal structure that resembled micro-communities of practice, which is central to knowledge management theories today is believed to evolve from highly efficient knowledge sharing and resembled the. (Thatchenkery & Chowdhry 2007) In the following ages, Aristotle emphases collaboration as a basic component of knowledge management and promoted creating research teams to deal with subjects of exceptional difficulty. Further, in the Middle Ages Irish monastic communities became powerful knowledge hub introducing knowledge codification and subsequent artifacts exchange, as well as one-on-one teaching sessions. (Paul McGrath 2005) Another important feature was ‘networked’ organizational model that differs radically from traditional hierarchical one. The networking took place at different levels, internally and externally, so it was not only a complex but an open system as well. Later, it was universities where full-featured communities of practice emerged remaining commitment to inclusiveness, discussion-based knowledge fostering model, and networking as a key progress enabler.

Unlike search for truth and knowledge unity that prevails during the Enlightenment, the modernist system was geared to obtain the best equation between the input and output and knowledge was no exception. (Lyotard, 1984). It got recon as a productive force for capital, thus the principles of efficiency and performativity came into play. Lyotard calls it «knowledge commodification» referring to fact that knowledge was treated as an asset, which should be leveraged to nourish human capital. Knowledge management (KM) evolved as “the formalised approach of managing the creation, transfer, retention and utilisation of an enterprise's explicit and tacit knowledge assets” (Cepeda and Vera 2007, p. 427)

Nowadays, innovation takes place within networks of organizations rather than individual firms resulting in cross-boundary knowledge flows appearing to be crucial requirement for such network-based innovations to come to live. (Man et al. 2008) Hence, so-called knowledge ecosystems emerge as a subproduct of business or innovation ecosystems, and partnership. The definition of knowledge ecosystem is assumed in agreement with Yoo et al. and implies co-evolving decentralized sociotechnical system that demonstrates synergy effects in terms of output, meaning generic knowledge production (Yoo et al. 2012), hence, thematic foci for research on the topic is output co-creation (Llewellyn & Erkko 2020). Ecosystems of this kind are aimed at facilitating incorporation of advances in research knowledge into products and services development or delivery. And according to Llewellyn & Erkko OS projects communities can be referred as an example of such an ecosystem. (Llewellyn & Erkko 2020).

### Modern concepts and framework

Due to increasing development pace and complexity of business environment tendency to reclaim appreciation in managing knowledge has been observed since 1980s. Appreciative Inquiry approach originating in the Case Western Reserve University can be regarded as a first precursor. (Cooperrider & Srivastva 1987) This method suggests facilitating discovery of existing assets and opportunities within teams with subsequent collective work toward search and implementing strategies for improvement by means of intense mutual communication. It covers organizational development in general with no particular emphasis on knowledge management. In the end of 1990s Grant suggested knowledge-based view of the firm approach that emphasis coordinating knowledge integration by strategic goal alignment in order to ensure that the knowledge was valued. (Grant 1996) The way the knowledge transmitted and received and the human-generated impetus for knowledge initiation and expansion got attention. Following, Thatchenkery used some of the Appreciative Inquiry principles for KM as basic for a new methodology called Appreciative Sharing of Knowledge (ASK). (Thatchenkery 2005) The ASK method provides a set of tools for acknowledging and improvement of already existing processes that are used for sharing “know how”. Thus, it does not imply building KM from scratch or changing people behavior. Neither it requires introduction of complex IT systems. that might have been beneficial at that moment yet seems to impose certain limitations nowadays.

Nonaka and Konno present more concrete theoretical framework addressing knowledge conversion process by which they undermine evolution of knowledge from tacit to explicit back to tacit, or vice versa. The concept is called SECI Model and identifies four types of knowledge combination and transfer, enabling knowledge sharing in the organization. The picture bellow depicts the cycle and provides examples of step-corresponding activities.

The model aims to explain the natural mechanism of knowledge creation and a way the process could be managed.

Each step includes an act of knowledge transfer that stands for communication with primary source of knowledge. (Foster et al. 2018) The model aims to explain the natural mechanism of knowledge creation and a way the process could be managed. Hence the underling mechanics of the process comes on the first place. Similar approach was proposed by Grover and Davenport defining knowledge-creating process steps to be generation, codification, transfer, and realisation. (Grover and Davenport 2001) Nevertheless, the latter is linear as opposed by SECI-cycle that obeys continuous-learning and improvement paradigm.

Diagram, schematic

Description automatically generated

*Figure 2 - Knowledge socialization – the spiral evolution of knowledge conversion and self-transcending process (Nonaka and Konno, 1998)*

In fact, despite attention drawn to knowledge formation, the mechanisms by which it is governed a little while ago have been largely ignored. (Heiman et al. 2009) Organizational and managerial theories emphasized exchange, knowledge transfer explicitly driving production efficiency, meanwhile governing the process of creating new knowledge met limited concern. It is knowledge governance that explains the mechanisms associated with knowledge creation, transfer, sharing and integration, inclusive of both process internality and externality. (Grandori 2001; Grandori 2009; Pemsel & Müller 2012). The mechanisms can be formal (explicit) and informal (tacit). (Foss, 2007; Foss & Michailova, 2009) Essential step to fulfil the gap was changing the unit of analyses from an object of action to its subject. In the mid 2000th Nickerson and Zenger's proposed PFPS that states for problem finding and problem solving perspective. (Nickerson & Zenger's 2004) Just as the name implies, the method is structured around issues identification and treatment. PFPS process is believed to be value creation enabler as it covers both search for prospects for improvement and its implementation.

To summarize, knowledge is a primary resource which has been involved during all the way humanity made up the Maslow hierarchy of needs over centuries. It has been always recognized as vital resource for a community thriving, thus, applied nature KM made it inevitable resulting in a rich experience and practice. Nevertheless, it is KG which deals with explanation of knowledge creation and distribution mechanisms that has been recently getting more and more attention. The study encompasses variety of field organizational, social, and business studies, as well as network and complex system analysis, so that describe a process that is believed to display synergy thanks to cross-fertilization. For this reason, the notion of knowledge ecosystem suggesting all associated with ecosystem concept features emerged with OS community named as one of examples.

## OS project community

OS projects are people oriented. Madey defines the OS movement as a complex, self-organizing system that consists of numerous locally interacting elements (Madey 2004). The system itself is an example of a complex evolving network and developers/users are nodes in the network. (Xu & Madey 2004) In this section, I first cover the semantic and business matters and then pass to the analytical concerns.

### Community as a major OS project success factor

Semantically, one can observe collaborative social network that emerges based on mutual interaction of its constituent OS product consumers that include both technical and non-technical end-users. (Fitzerald, 2004; Nichols & Twidale, 2003). OS community has transient nature, with the participants being mostly volunteers and occasionally paid contributors (for instance in organization-adopted or even organization-led OS projects, such as Apache Ignite (Sber Technology)[[7]](#footnote-8) and CatBoost (Yandex)[[8]](#footnote-9) respectively. Most prominent reasons for volunteering in OS projects are new skills acquisition, self-realization, tools customization and affirming reputation by means of code contribution or leading pet projects. (Crowston 2011) This way developers can simultaneously enhance own UX and enrich their CV benefiting career opportunities. Major OS projects are seen as a valuable source of expertise[[9]](#footnote-10), and its proactive collaborative culture is believed to be worth spreading[[10]](#footnote-11). Nevertheless, despite obvious strengths, there are certain peculiarities that makes OS culture adoption challenging, for instance, diversity of expertise result in misunderstanding, and lack of robust central authority leads to poor coordination. These features bring us back to the ecosystem concept that is briefly mentioned in the section concerned with knowledge, it is time to take a closer look on it.

Indeed, open-source project can be recon an ecosystem, since it meets all the definitive criteria proposed by Llewellyn & Erkko, namely participants heterogeneity and intercedence, decentralized governance, and non-linear generic output. As stated in Bass *et al.* “*most OSS projects do not have a single identifiable architect, and the architecture is typically the shared responsibility of the group of committers*”. In fact, OSP can be qualified not only as knowledge ecosystem, but an innovation ecosystem as well, where priority is given to applied knowledge and strategic advances that ensure competitive advantage acquisition sustainability. (Llewellyn & Erkko 2020) Llewellyn & Erkko subdivide innovation-ecosystem according to foci:

1. Business ecosystems - focus on stakeholder involvement.
2. Modular ecosystems - focus on generic value offering that is targeted at particular audience.
3. Platform ecosystems - focus on interdependences coordination.

I reckon that OS is best described by platform ecosystem concept, that feature strong technological interdependence and generally employ technological architectures and platforms as their co-alignment structure. The platform usually appears to be Version Control System (VCS) that is available through internet and offers additional project-management oriented tools such as task-tracking, communication-oriented sections, etc... Well-known examples include GitLab, GitHub and SourceForge. Concerning PFPS approach, key activities take place via special tool provided by the platform that serve variety of processes ranging from customer support to change and release management that are an essential part of OSS routine. (Fitzerald, 2004), (Krogh et al., 2003) So far for the sake of revealing the nature of associated activities researchers have mainly focused on mailing lists. Mailing lists are discussion-based and allow for solving stack-holder identified problems that reflet identified gaps between present and desirable situation either in terms lack of competency or hidden value. Participatory culture prevails enabling sharing of ideas and knowledge, and social media (represented by online collaboration platforms) plays an essential role in this culture. Research conducted by Jensen et al. shows that timely response is positively correlated with likelihood of further participation in the OSS community (Jensen et al. 2011). Thus, knowledge spread is crucial for OS project thriving.

Documentation is a commonly accepted source of knowledge, however, findings of Ding at al. indicate that unlike organization-led projects it has not been widely used in OS development. The former study shows that is true even concerning such a fundamental area as software architecture (SA). Though, the fact is not due to lack of necessity, as the likelihood that an OSS project will SA increases when more developers get involved in the project, which suggests that SA documentation appears to be more useful to OSS projects when there are more developers. (Ding at al. 2014) Indeed, decentralization as basic feature of OSS projects entails impossibility of prescribing precise rules that proposes significant challenge. In other words, owning to either community failure or OS phenomena specificity, formalized knowledge sharing mechanisms in this case appear to be of limited robustness. Hence informal peer-to-peer knowledge communication should be let first. One form of those is called meaningful interaction. In psychology, the notion is used to define communications that are intended to meet the needs of others. (Kimmerle et al., 2010; Carpendale & Müller 2014). Touching on the format that the latter take, research by Yutaka et al. shows that in OS participants mainly communicate via electronic media (e.g., forum, mailing lists, etc..) rather than face-to-face. (Yutaka et al. 2000) In the same vein, Red Hat's Tom Callaway names absence of communication tools as one top reason OS projects fail, along with newcomer-unfriendliness and loss of core members. Moving on, Sowe et al. suggest distinguishing between OS project participants into two categories:

1. knowledge provider - one helping others on various issues related to software development and exploitation.
2. knowledge seeker - one in search for assistance on software development and usage, like code compilation, details configuration, etc.

Each member of community may occupy any of those position, simultaneously or changing over time. (Sowe et al. 2007) Nevertheless, it is still possible to distinguish a group of key knowledge providers that are the most experienced and skilled participants and occupy high positions in OS hierarchy, namely core contributors those and maintainers who possess ability to decide on the project evolution, e.g. accepting or denying new features and contributions, project roadmap development and ect.. The group is a tiny fraction of the community, yet their impact is ccucial for knowledge creation and sharing activities. Research held by Lakhani and Hippel on Apache ‘field support system’ shows that about 2% of the knowledge providers oversee nearly one half of the answers to questions posted. (Lakhani & Hippel 2003)

### Open-Source Economics

Conventionally, economic concerns allocation of limited resources. Most of times they refer to money or at least somewhat estimable in monetary terms yet touching on OS projects traditional compensation mean are of limited applicability given prevalence of volunteers. Mike McQuaid, Principal Engineer and Homebrew Project, claims that money inflow cannot help OS project health unless proper maintaining activity, such as open issues monitoring, pull requests review, etc.. takes place and is properly coordinated[[11]](#footnote-12). According to McQuaid, all the listed duties rise on the shoulders on the participants group that mostly consists of maintainers and core contributors, otherwise, those we have previously defined to be key knowledge providers and pointed out the negligibility of the group share in the whole community. Thus, the challenge concerned particularly with limited labor resource allocation arises, meaning skilled people and their time. In the same talk McQuaid emphasize the violability of sufficiency and sustainability of participance of the kind resource for community thriving and project overall health. Overload leads to these people burnout and designation, and designation of key knowledge provider result in crucial knowledge loss. (Rashid et al. 2017) Even worse, the process of knowledge drain is highly likely to be positive feedback process, since findings by Lakhani and Hippel indicate that most productive knowledge providers appear to be most enthusiastic knowledge seekers, and it is the tradeoff that encourage involvement in learning activities on voluntary basis.(Lakhani & Hippel 2003) Hence, the question is whether there is lack of maintainers resource of it is just poorly allocated. Concerning individuals, the time they are available for is not flexible, as significant part of them contribute to OS project for hobby and are not ready to work in expanse of their core occupation, never to mention to quit that. As a result, two possible ways of fighting the bottleneck are usage of latest smart helpers embedded in the platform the project is based on (such as Issue forms for customer support tickets unification and validation) and facilitation increase in number of maintainers if the project owners can afford it.

### Statistical properties of OS community

Most prominent project participants can be identified by means of social network analysis (SNA). In fact, SNA has been widely applied to OS community related research. (eg. Madey 2002, Jin et al. 2005, Sowe et al. 2007) Unlike traditional social networks there is generally no personal chats embedded in VCS platforms, for this reason, the ties between OS community members are established thanks to reciprocal involvement in a certain activity. Hence, concerning the network that represents OS community two kinds of nodes naturally emerge. Namely they are agents (participants - users/developers) and artifacts (the integrating media, such as repository, which people in question follow; issues they track, or mailing lists). I would like to start with the part that is concerned with humans and then proceed toward artifacts-related connections.

Regarding the community, underling structure of OS project resembles scale free network meaning that its degree distribution obeys power law. (Xu & Madey 2004) In accordance with Barabasi and Albert (Barabasi 1999), such a network possesses two properties:

1. It grows by the sequential adding of new nodes - there are new commers in OS projects.
2. It displays “richer gets richer” phenomenon, in other words, probability of two that nodes get connected increase with the nodes' degree - in OS, developers are more likely to join top-rated projects.

Moving on, Sowe et al. investigate mailing lists when urging for understanding knowledge sharing activities in F/OSS projects and the potential way the latter may affect development process. To ensure the sampling was large enough research chosen Debian project and combined data derived from two kinds of mailing lists, that are the Developer and User mailing lists (DMs and UMs). The reason is that discussion held through DMs covers exclusively core development processes and involve only maintainers and possibly most knowledgeable contributors, thus, the associated output in terms of communication artifacts has been shown to be of limited volume. (Healy and Schussman 2003; Kim 2003) On the contrary, UMs are used by those needing assistance that suggests little to no requirements for one’s competency never to mention discussion topics are literally endless, hence, there plenty of artifacts of this kind. Knowledge sharing is assumed to take place once two or more participants exchange messages. They propose the following model that can be aligned with SECI model introduced in section 2.1.2.:

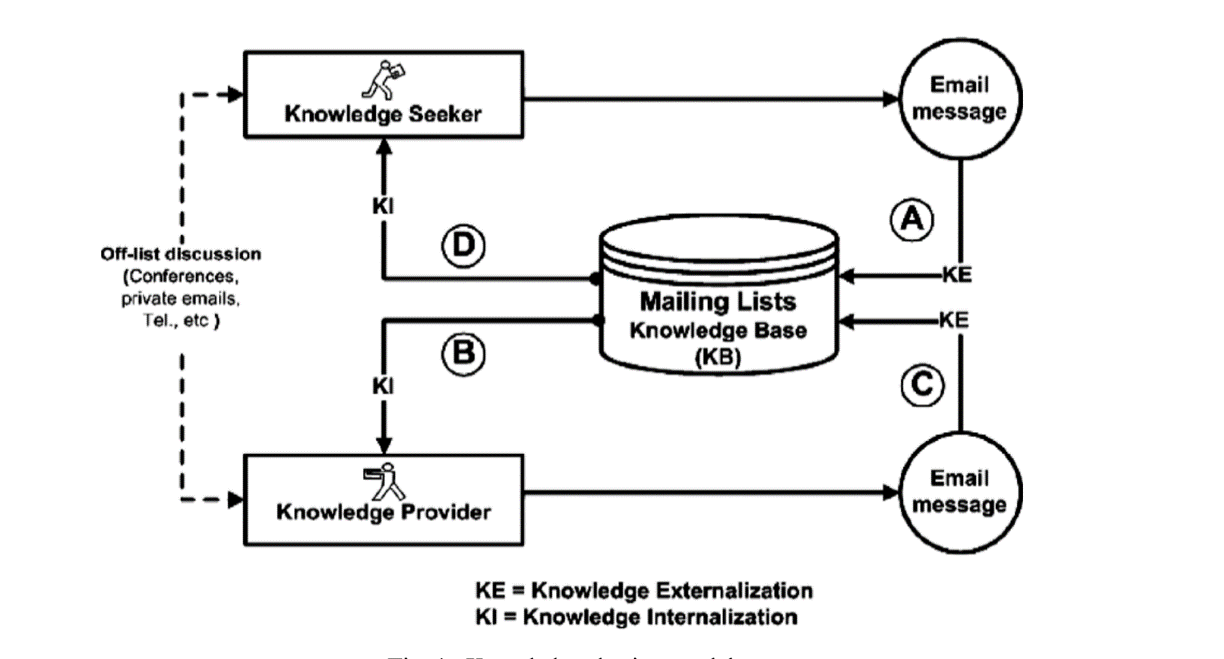


Figure 3 - Knowledge sharing model. (Sowe et al. 2007)

In short, there are two possible patterns depending on whether the problem has been encountered before or not. In the first case, it is assumed to have been captured in the knowledge base (represented by collection of threaded discussions), which stores externalized knowledge resulting from interaction between list participants, hence one can directly access it. In the second case, when the problem has not been addressed so far, knowledge seeker makes a post on his question, thus provoking ideas exchanges leading to externalisation of knowledge held by the ideas owners. At the same time knowledge is also internalised as participants consume information from the discussion. The is also a chance that knowledge seeker can communicate with knowledge providers outside the platform. The letters on the diagram are used to denote the sequence of actions taken by knowledge providers and knowledge seekers.

These is no proven way for measuring knowledge especially when it comes to tacit one. (Atreyi and Bernard, 2004) Sowe el al. research group suggests following metrics:

1. the total number of posts that constitute for knowledge externalized by knowledge seekers.
2. the total number of replies that correspond knowledge externalized by knowledge providers.

To ensure that mutual information exchange takes place OS community members with zero number of posts or replies were removed from data, as well as those with invalid email addresses. Moreover, posts with unfilled subject or system-generated, such those concerning s cancellation of subscriptions are not included.

|  |  |
| --- | --- |
|  |  |
| *Comment: nreplies and nposts variables are used to denote the (whose subject starts with “Re:”) and initial requests (no “Re:” in the subject).* | |
| Chart, line chart  Description automatically generated  Figure 4 - Power-law distribution showing linear relation between participants’ posts and replies and their ranks in the two lists. (Sowe at el. 2007) | Diagram  Description automatically generated with low confidence  Figure 5 - Power-law distribution showing linear relation between participants’ posts and replies and their ranks in the two lists. (Sowe at el. 2007) |

The study shows that distribution of the posts and replies in both lists overall obeys powerlaw just as one should expect from self-organizing system. However, the research demonstrates that the distribution could be more precisely described using so-called “fractal cubic distributions” implying certain fractal (self-similarity) properties, hence there are ground for expecting higher level of complexity.

Wrapping up, knowledge creation, that powers OS projects development, deals with natural synergetic process evolving within an open self-learning and self-organizing system. Quantitively research conducted by Sowe et al. shows that knowledge creation activities can be described in terms of powerlaw-like distribution, Still the same study indicates that even a better fit can be achieved by introducing another kind of distribution which displays fractality. In fact, the observables they provide can be referred as purely “spatial” ones, as they are represented by a pattern on graph and do not reflect the evolution time. What I mean is that the number of connections as a function is a single-variable one which depends on the node choice with no regard to time axis. Asawhole, given features of the phenomena in question suggest self-organized criticality, meanwhile, temporal fractality along with some other definitive characteristics are to be verified.

## The SOC concept and defining features

### Definition and history

For the sake of simplicity, let us start with a real-life example. Consider a sandpile located on a flat tray with no boards. Let the sandpile be stable, meaning that no sand is moving in the moment. That is the initial setting. Now, one starts adding sand grains one by one at random places. The single act of adding a gran is called “toppling”. So far there is nothing going on with the pile. Eventually, the slope becomes too steep resulting into a sapping. Once one continues long enough the sappings become numerous and varied in size. More importantly still, it that a single toppling is able to provoke a sequence of sappings, and even including disproportionally big ones. Still, the sequence is not going on forever thanks to the tray being borderless, as time to time grains fall down and the system becomes stable again till the sandman’s effort get it to the critical point again.

Academically, self-organized criticality takes place when interacting elements organize themselves towards a critical state with no fine intervention of external inputs and regardless of the initial conditions. Due to *perpetuum mobile* (a system needing no external energy input to function) being absent in nature, there is no way to eliminate some intervention completely. Notably still, the sandman adds grains with no accurate calculation in mind, it is the slope that provokes sappings and not the actions directly. The input can be further generalized to the abstract notion of external energy that occurs randomly and leads to local disturbances. The disturbances, in their turn, lead to a little or largely amplified output revealing non-linear nature of the system. As the tray is borderless and sand grains are free to fall, or in other words thew quantum of energy can leave the system, the system is also dissipative. Let now assumed that there is constant toppling, then the sandpile grows with incidents that are existent grains movement, otherwise, one can say that it grows in a stablale manner. Periodically, the relentless topplings lead to a sequence of avalanches of random size, herewith no correlation with the input rate of sand can be observed. Wherever, such situation takes place, one deals with the critical state. For SOC, the critical point is an attractor of the system in question.

The first example of a system displaying SOC phenomena is BTW model. (Bak et al. 1987) The pioneering papers by Bak et al. (1986) and Katz (1986) were followed by extensive studies in variety of fields. It is diverse branches of physics that can be considered as SOC primary domain, yet recent applications increasingly address biology, natural hazards, and human activities.

A key concept associated with SOC is “avalanching” that covers interaction of a large number of elements that takes place over a short period of time in a dissipative system. In essence, it is a just an abstract notion for a complete series of sappings. (Aschwanden ed., Chapter 1 by Norma B. Crosby)

The system is continuously driven by random input towards critical state in which a minor event starts a chain reaction engaging arbitrary number of elements. For illustration based on sandpile example consult see figures 6-9. Let assume that the orange sapping was the final one for that period, hence the avalanche incudes just two sappings those are the pink and the orange one. The critical state is characterized by powerlaw frequency distributionof characterizing parameters.

|  |  |
| --- | --- |
| A picture containing text  Description automatically generated  Figure 6 - Step one. Random grain input. | A picture containing text  Description automatically generated  Figure 7 - Step 2. The input drives first "pink" sapping. |
| A close-up of a shoe  Description automatically generated with medium confidence  Figure 8 - The pink sapping grains appear to be an input for the second sub-sandpile. | A close-up of a shoe  Description automatically generated with medium confidence  Figure 9 - The pink input leads to critical slope for the second sub-sandpile and "orange" sappings starts. |

If any input is interrupted until the resulting avalanche dies out, the model is called a *“stop-and-go”*, whereas providing driving continues rregardless one deals with *running sandpile*. The first case relates to another important concept that is “slowly driven” system. That means that “so that the characteristic time scale of the driving does not interfere with the internal, fast time scale of the relaxation.” (Watkins et al. 2015, p. 22) In other words, the sandman waits until the avalanche vanishes (relaxation period) before start adding grains again, moreover duration of any given avalanche is minor in comparison to the period that is needed for the system to get to the critical state. In its turn, “self-organizing” means that no matter what the input is, the system evolves towards invariant endstate that implies characterizing exponent for the powerlaw distribution stays the same. All together those features constitute self-organized critically.

Touching on rigorous definition, SOC can be validated through investigation of proper event statistics, which include time and size scales of avalanches. One should expect scale-free powerlaw distributions of time and size scales, as well as some sort of randomness touching on waiting times. (See figure 10 for illustration).

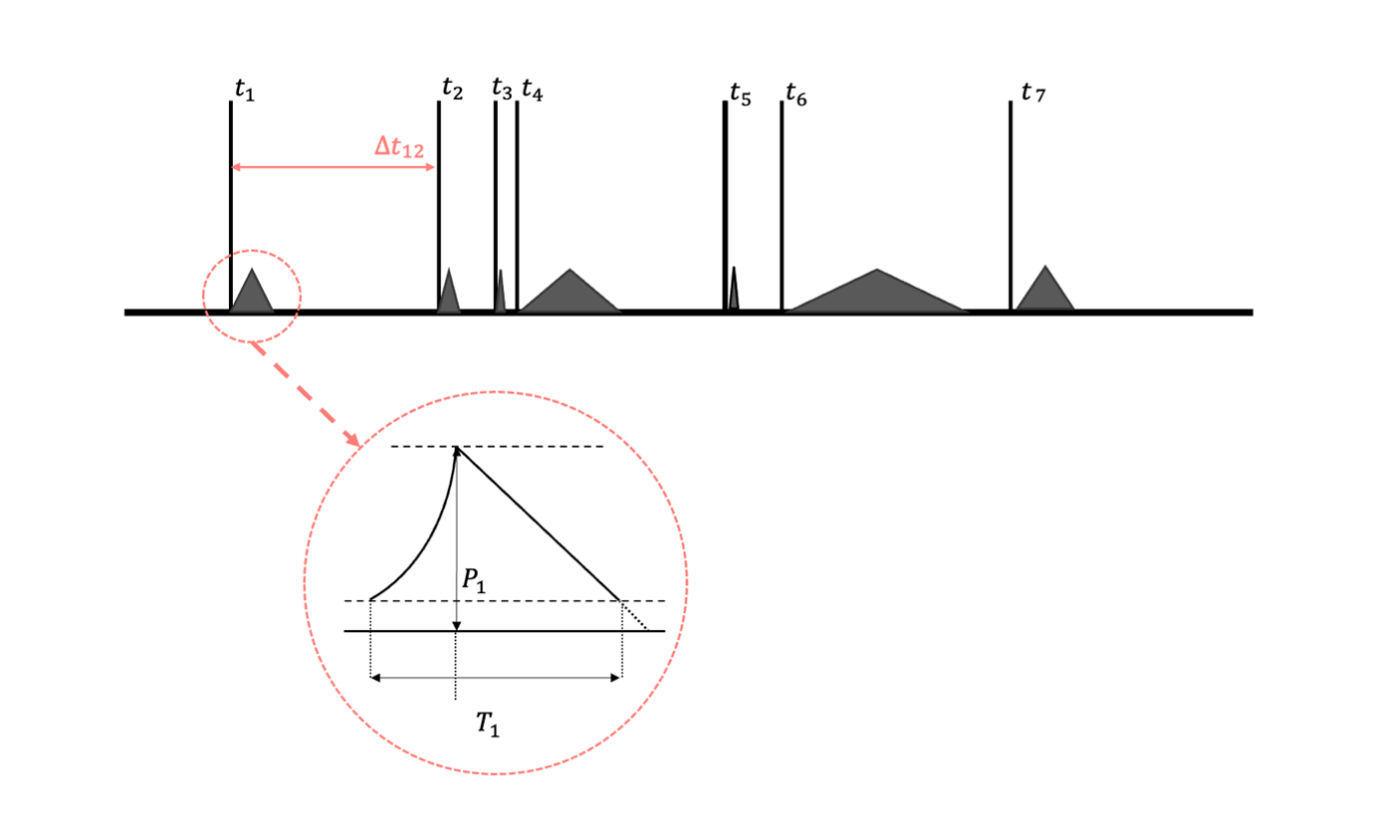


Figure 10 - and denote events (avalanches) and waiting times between subsequent events respectively. Each event consists of growth phase (when the chain reaction starts and accelerates), which is followed by peak energy release – (e.g. avalanche size/area), and finishes with decay or relaxation phase when chain reaction weakens. is event duration. (Adopted from Aschwanden 2011.)

Namely, three distinguishing features of SOC can be identified (Aschwanden 2011):

1. Statistical Independence

This one refers to avalanches that occur in a SOC system and are statistically independent in space or time. As a practical matter, one should test whether waiting time distributions is consistent with a stationary or non-stationary Poisson process to ensure statistical independency. Indeed, the classical sandpile model suggests no correlation between the input and the output, thus, the statistics of waiting times between subsequent avalanches should be random. Waiting time is the time interval between subsequent events and depends on the definition of the event times ti.

1. Nonlinear Coherent Growth

To put it simply, at a critical state once chain reaction starts it does not weaken until peek avalanche size is reached. The growth itself is nonlinear and exhibits an exponential-like or multiplicative time profile for coherent processes.

1. Random Duration of Rise Times

The randomness of rise times manifests itself through the statistical distributions of the latter being consistent with binomial, Poissonian or exponential functions.

### SOC models

Overall, models of SOC can be divided into two categories that analytical and numerical classes, the latter are mostly presented by cellular automaton (CA) (Aschwanden 2011). Due to non-linearity the complex macroscopic structure cannot be analytically derived from the microscopic states of lattice cells that deals with. Thus, analytical models can at best provide a simplified approximation of the SOC dynamics, which are aimed to be consistent with the observed frequency distributions. Nevertheless, such models are used to make quantitative predictions of the powerlaw slopes and scaling laws. Touching on CA, being cellular refers to the fact that the model is discrete concerning space, while autonomy implies that despite openness to external drivers, e.g. grain inputs, the system state changes with to intrinsic process and in accordance to internally tuned parameters.

The CA model includes following elements:

1. S-dimensional regular lattice grid.
2. Set of «grains» number corresponding to each lattice node.
3. – value for critical threshold, or a maximum number of grains that a node can hold.
4. Random input in space and time
5. Table

   Description automatically generated with low confidenceMathematical re-distribution rule that defines the quantity (or ratio) in which grains are transferred to neighboring nodes once the critical threshold for a given node is exceeded (). Traditionally, redistributions occur with neighboring cells. (See figure 10 for example for )

Figure 11 - Basic re-distribution rule application.

The BTW model was one-dimensional of the those described above. The envelope of all unstable cells () is regarded as the area of the avalanche event and the time duration stands for time steps that the avalanche continues. Subsequent enhancements include replacement of the regular lattice with arbitrary graphs (e.g. Dhar 1990), introduction of stochasticity in the redistribution rules (Lefebvre & Mukamel 1987), and development of continuous version. (Kalinin et al. 2018).

## Related frameworks

Powerlaw behavior is fairly common for both natural and social systems statistics. Beside SOC it may originate in variety of phenomena, for this reason, in this chapter I provide brief descriptions of those emphasizing discriminating features in order to justify choice of SOC as primary concept.

### Self-Organization without Criticality

Self-organization (SO) is frequently associated with emergence theory, namely geometric patterns evolving from mutual interaction of their element in the absence of external coordination. Some examples of SO include regular pattern of the desert sand dunes (Bangnold 2005), and geometric patterns in biology, such as the animal skin or formation flight of birds (Camazine et at. 2001).

SO emerges from non-equilibrium processes governed by long-range interactions and covers geometric patterns that are to a certain extent stable over long periods. The difference between SOC and SO mainly lays in the scope of interactions and scaling. Firstly, in case of SOC events are triggered by localized disturbance individually and independently, whereas SO implies system-wide interaction. Secondly, for SO exists so-called preferred size scale, such as common width of sand dune wavy ripples. On the contrary, SOC generates scale-free avalanche distribution. Thirdly, SO produces spatial patterns, while dynamical events in space and time are peculiar to SOC.

### Brownian Motion and Classical Diffusion

Originating in classical physics this concept referrers to random motion of atoms or molecules in gas or liquid. The key difference is that the Brownian motion (or alike classical diffusion process) lacks self-tuning aspect being unstoppable. There is no way to distinguish between events and, consequently, derive frequency occurrence distribution.

### Percolation

According to Aschwanden, percolation covers a transport process that can characterized by propagation probability of nearest-neighbor elements. Knowledge flows have been examined from the perspective of percolation model (Jigsaw) providing an insight on more efficient community structure. Still the mechanics of concomitant knowledge development process is out of scope that paper, arguably due to lack of means regarding absence of self-tuning feature of percolation model, whereon the fact that the outcome depends sorely on initial conditions.

### Chaotic Systems

Systems of the kind also display fractality, intermittency, and occasionally even scale-free distribution regarding chaotic fluctuations. Nevertheless, the chaotic aspect of such systems’ behavior is in fact limited to extreme initial condition sensitivity, while the overall evolution of system remains deterministic and evolves from system-wide dynamics. On the contrary, in case of SOC avalanches result from nearest-neighbor interaction and are assumed to be triggered statistically independently. Furthermore, even though critical points from where on chaotic behavior became evident are commonly present, it is not a driving mechanism that tunes the system to the point.

### Forced Self-Organized Criticality (FSOC)

Extensive usage of the FSOC lies in the field of astrophysics (eg. Chang 1992, Lue et al. 2000, Kozlov et al. 2004). These two phenomena are alike in terms of observables as they both suggest scale-free avalanche-size distribution, fractality, intermittency, statistical independents of events and critical threshold. The division derives from the driving force of instabilities that appear to be random localized disturbances and long-range coupling.

### Network Systems

The original cellular-automaton model of SOC is based on a regular lattice with neighboring cells being subjected to re-distribution rule. The obvious difference between conventional SOC models and network systems is lattice irregularity and long-range coupling. In addition to this, since emergence of hubs is just natural the probability of nearest-neighbor interaction for given node significantly varies, meanwhile regarding lattice points it is statistically identical.

Nethertheless, many network-related phenomena have been proven to display SOC behavior despite of the presence of long-ranged connections. Thus, usage of network structure instead of the lattice is fairly common enhancement in SOC studies, especially then it comes to human-related subject areas, such as urban studies (eg. Zannette 2007), financial markets (eg Feigenbaum 2003), social networks (eg. Newman et al. 2002), etc. Moreover, concerning SOC as a part of complexity theory NS provides an essential assistance in terms of reflecting attributes of the original system and its dynamics by mapping the latter onto a network (usually represented by graph), so that the design of a graph can be interpreted in regard of the fundamental dynamics and interactions between the constitutive units of the system. (Tadic ́ 2019) Thus, mathematical construct and the subject area are bridged. Last but not the least, such mapping enables usage of graph theory for objective analysis. (Dorogovtsev 2010, Bollobas 1998). In particular, network analysis has been westly applied to social studies touching on knowledge creation and spread. (eg. Dankulov et al. 2015, Andjelković et al. 2016, Swan et al. 1999)

To summarize, it can be observed, that most statistical aspects are shared between SOC and SOC-related models. Thus, it is self-tuning mechanism along with the system characteristics, namely, openness, dissipation, and non-linearity that can be recon as the key discriminating feature for SOC.

## SOC and knowledge flows

SOC applications to knowledge domain are limited. (Tadić et al. 2017) Still there are a few papers and in this section, I would like to present two of them chosen by business process similarity, namely those investigating knowledge creation process accompanying scientific collaboration and Q&A site routine.

Tadic ́ et al. study Q&A forum dynamics. (Tadić et al. 2017) The propose an agent model build on bipartite network that consists of human agents (forum members) and artifacts which incapsulates questions and associated answers. The agent actions that are posting an answer or a question represent toppling. And an avalanche is defined as the activity periods withing those the total number of actions doesn’t drop to the zero line. The model that is used by Tadic ́ et al. combine features of FSOC and NS. Stated assumptions include arrival of new agents as a sub-driver on the top of Q&A activity representing toppling, hence suggesting FSOC. Touching on NS attributes, not only agent interaction is not limited to nearest neighbor connected artifacts, but there is non-zero probability for a random artifact to be subjected to action as well as increased probability for an artifact connected to the agent neighbors to be chosen, therefore, long-range coupling and long-range connections can be observed. The research incorporates empirical data analysis and theoretical modelling examine in order to study the temporal correlations as well as the cognitive contents significance to the avalanches in knowledge creation process. It demonstrates that it is social dynamics combined with is an external driver, such as new users inflow that leads to avalanching behavior. Meanwhile, the cognitive content was shown to shape the structure of the developing knowledge network, they only affect the avalanche parameters values. On the top of Tadić at el. report correlations between knowledge sharing and fluctuations of the innovation rate. The latter is defined as new competency combinations resulting from agents that heterogeneous in terms of expertise participate in the same discussion.

Tadić et al. do not elaborate on the network structure, however there are papers that investigate mutual relation between the two scale-free degree distribution and similar distribution of events that characterize the associated system. (e.g. Bianconi & Marsili 2004, Paczuski & Hughes, 2004), Touching on knowledge, the relevant research has been conducted by Fronczak et al. (Fronczak et al. 2005). The study by Fronczak et al. is based on simulation within BTW-sandpile model that has been enhanced by replacing the regular lattice with scale-free network that corresponds to empirically obtained scientific community structure. The model implies that node and an edge represent a scientist and the ability to exchange ideas respectively. The “energy” (an abstraction for sand) is potential which is non-decreasing function and describes one’s ability to produce high-quality paper. Once the critical value of potential is reached, the paper is assumed to be published. The paper inspires other scientists, thus triggers other paper to get born, hence an avalanche occurs. Therefore, big avalanches result to research areas. Important assumption is still that the number of works one agent is capable to observe is limited, for this reason agents rewire from older studies in favor of the last and most interesting ones, reshaping the network. The paper concludes that the network structure and the social dynamics are capable to influence each other that, in its turn result into self-organization with equal scaling exponents for both avalanche distribution and degree distribution become identical. To be precise, τ = 2 was calculated. The researchers suggest that may be universal in “ ’in’ the structure network processes”. scientific collaborations. Yet, it does not coincide with estimates by Tadic ́ et al. who found both nodes (agents) degree distribution scaling exponent and those of avalanche size distribution to be well bellow 2.

To conclude, SOC-consistent statistics has been observed in social systems dealing with knowledge creation, yet the question on the universal model the process can be reduced remains open. Thus, for each case adjusted version should be proposed.

## Applications

The study is aimed to explain knowledge development process occurring withing OS project ecosystems in order to support evolving new organization or even governmental structure-led OS movement wave that pioneered in terms of tight bonding with corporate standards. Nevertheless, community is a special kind of voluntary labor force that has an enormous, but measureless potential for innovation. Obviously, no metric can be derived without process defined, yet little can be done providing the process is not described properly due to lack of understanding of the underlying activities. The next part of the study focuses on essentially on knowledge creation process description with usage of SOC-based theoretical framework.

# Chapter 2. Applications

## Theoretical framework

The final section of literature review builds around two papers that suggest different approaches to defining advances in knowledge development process. Both of them are in a way appropriate for OS community case corresponding to particular kinds of problems uncovered. To be precise, bug reports and customer support related requests are more like social endeavor taking place at Q&A site (Tadić et al. 2017), meanwhile feature suggestion and open discussions on the project prospects seem to sooner resemble research areas (Fronczak at al. 2005). The two approaches may be found contradicting as the key model element is either physical and coherent (research area) or an abstract aggregation of physical events (number of actions undertaken by agents in a period of time). For this reason, the methodologies may be found contradicting, still I suppose they may reflect the system complexity corresponding to different research scope, (Treur 2020) Indeed, the overall sum of actions encapsulates those within certain field of study. Thus, I would conduct two experiments corresponding to each system level.

As I have mentioned before knowledge ecosystem that evolve around OS projects are platform-based, hence it and the associated process reflect platform peculiarities and functionality. Thus, touching on data collection it is not only community and events, but also platform to be chosen mindfully. I will start the section with brief overview of most notable platforms to support the choice and subsequently proceed to events definition and data collection method as the latter are platform bounded.

### Platform choice

**SourceForge**

Existing since 1999 SourceForge is one of the oldest platforms for joint developments of OS projects. It has been widely used for scientific purposes when it comes to research on OS projects development flow and community dynamics (eg. Sowe et al 2007). Nevertheless, in mid 2010s it has lost most its users due to controversial monetization practices[[12]](#footnote-13). On the top that communication tools provided by the platform lack performability given currently achievable involvement level.

**BitBucket**

In 2010, the BitBucket project was acquired by Atlassian. Among the platform’s main advantages native integrations with Atlassian ecosystem applications, such as JIRA, Confluence, and Trello should be mentioned. However, in the context of the current study, this means that the desired communication tools are taken outside the platform. Many companies use Trello+Jira as a task tracking system, and Confluence serves as a knowledge base. It is also necessary to note the predominantly commercial nature of the platform. As a result, the BitBucket community is relatively small: as of April 2019, only 10 million users and about 28 million repositories were stated.[[13]](#footnote-14)

**GitLab**

Gitlab started as an open-source project, yet in in 2013 two product editions were released, namely Gitlab CE: Community Edition and Gitlab EE: Enterprise Edition. It was in 2014 when Gitlab EE became a commercial product, at the same time the functionality of the free version was severely restricted. Moreover, in August 2018, Gitlab moved from Microsoft Azure to Google Cloud Platform, which made the service unavailable in a number of countries due to limitations imposed by the Office of Foreign Assets Control of the United States. All combined, the changes impacted the users’ demand: The GitLab community is significantly less representative than those of GitHub. In general, I would say that GitLab seems to be more focused on professional users since it provides an expanded toolkit and services for release management, deployment, and monitoring of services. (In the commercial version), meanwhile, neither the educational component nor wider community accusation is among GitLab’s priorities.

**GitHub**

Github, Inc was founded in 2007 in San Francisco. In 2011, the platform surpassed SourceForge in the number of comments, and by the end of 2021 it had become the largest of the listed platforms with the most developed community of over 61 million repositories and 73 million active developer users. Previous year more than 16 million new users joined the platform.[[14]](#footnote-15) In July 2018, GitHub was announced to join Microsoft, and just two weeks later, free educational projects were launched ranging from the school level. Ever since Microsoft has declared its intention to support innovations aiming at developing a sustainable community around projects, and productive collaboration of developers and users, looking for new forms of interaction.

In view of the foregoing, GitHub was chosen as basic platform that a knowledge ecosystem to be studied has grown on. Firstly, the size and maturity of the community indicate the possibility of obtaining a representative sample. Secondly, the proposed communication and learning tools under the patronage of Microsoft have received significant development and reflect modern practices, which corresponds to the subject of the student's thesis. Thirdly, from a practical point, GitHub offers robust API for data set formation.

### Knowledge development process

It is necessary to take into account the way information is communicated and the methods of communication in OS project communities, since joint development platforms generally and GitHub in particular do not have built-in messengers. Hence, instead of one-on-one communication, one can only talk about open discussions, the number of participants of which is not limited. Meanwhile, it is the lack of privacy of the resulting dialogues that makes it possible to form a knowledge base accessible to all project participants, as well as to involve a wide range of people with diverse expertise in the discussion. This part will describe the types of artifacts that result from using the built-in communication tools on GitHub. The majority discussion around OS projects set around change requests, bug reports, brainstorming, and user support. GitHub offers several options for communication purposes as a part of its infrastructure. The choice of a specific one depends on the use cases, community code of conduct, and agent preferences. Next, we will consider each of the types in details, indicating its features and use cases offered by GitHub officials[[15]](#footnote-16).

**GitHub Issues**

Designed for targeted rather than general discussion of project details, such as bugs reporting, planning changes, suggesting improvements, and user feedback. This type of artifacts directly stores the knowledge that belongs to the OS project strategic resources, as it helps create additional value for users by means of value co-creation and insights generation. Issues relate to a specific repository and can be uniquely attributed to the authoring project participant. Typical use cases include tracking change requests and tasks tracking, bag reporting, getting feedback on a specific feature, and Q&A.

**Pull requests**

Pull request are often result from issue. They stand for a request to make specific changes to the project code, as well as implement the code-review mechanism. Pull requests also relate to a specific repository. In other words, it deals with modification of the code base or documentation. For this reason, the usage of this artifact type is limited to the most qualified and involved project members. Scenarios for using Pull requests are error correction, source code modification, issue request fulfillment, and commenting on changes made or imposed.

**GitHub Discussions**

A relatively new feature on GitHub, proposed as a beta in 2020 and released in 2021. The Discussions section provides familiar forum-like interface and are designed to support discussions of a wide range of ideas, including the project and the accompanying community identity formation, newcomers’ adaptation facilitation both in terms of technical skills and culture-fit. Discussions can be linked to several repositories, thereby helping to form a true ecosystems of OS projects with shared knowledge base. As stated in GitHub wiki this artefact can be used for discussion of issues, the scope of which is not limited to one repository, communication of news, project plans, announcements, and open discussions.

Studying free knowledge flow between community members, it is reasonable to abandon the consideration of pull requests because of both narrow scope the source covers and close deeper connection with innovative component of the community's activities implying applied knowledge. On the contrary, discussions seem to be most unstructured and have the potential to involve the widest range of stakeholders thanks to being least demanding in terms of technical. However, the novelty of this tool does put under threat sufficiency and representativity of the available data since both the size of the sample and the persistence of user behavior patterns are doubted. Thus, the choice for activity to analyze was made in favor of GitHub issues as a comprehensive knowledge carrier artifact that combines strategic significance and robust semantic coverage.

Touching on the limitations of the study by Sowe et al. which has been introduced in the literature review section, the authors recognize that data cleaning procedure that has led to removal about half and two third of the participants from the UL and the DL respectively may affect the findings, namely result in better fit of cubic-fractal distribution rather than powerlaw one. (Sowe et al. 2007) In addition to this, as far as I concern, another problem might have been lack of tacit knowledge volume incorporation. Perhaps that was due to data collected that does not allow for such measurements. Otherwise, that could be a conscious decision as the violability of attempts to estimate that kind of knowledge is doubted. (Atreyi and Bernard, 2004) Nevertheless, the case when the desired knowledge piece has already been externalized to the knowledge base has not been uncounted. Meanwhile, I suppose that the decision may have been erroneous, hence I would like to present include this type of data points, too. The process is schematically presented on Figure 13.

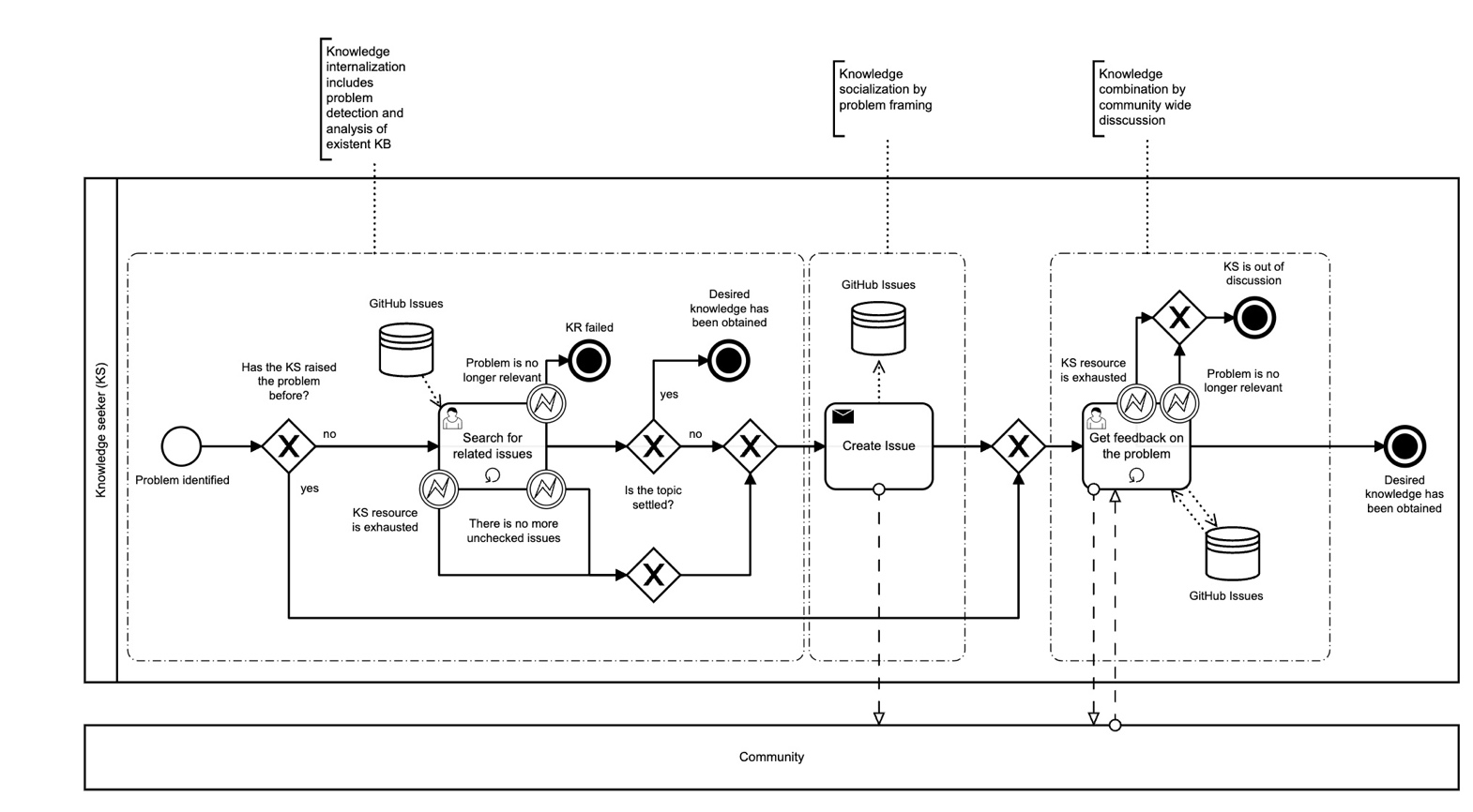
Building on SECI model, I would like to stress necessary changes that should be applied. The platform serving as mediator in most cases excludes external personal communication, however, the semantic load, which is the socialization of the request for knowledge, can still be implemented. If the problem in question has already occurred before, then the act of codification has already been performed, hence one may access the desired knowledge piece through the iterating on existing issues (fortunately, the fact of viewing the issue is traceable in the data provided by the GitHub Archive). Otherwise, a new issue should be created, so that subsequently a response could be submitted. Once an issue is created or found knowledge can be internalized from the knowledge base, either by knowledge-seeker (according to the original request from the previously existing issue) or potential knowledge provider (learning from the newly created issue about the gap in the knowledge base). Meanwhile, the issue owners or other community members can join the discussion at any point, which marks the combination stage. It should be mentioned that codification occurs at each stage. Thus, unlike the in case of original SECI-cycle, it does not appear to be as a separate element but accompanies each step.

Figure 12 - Knowledge development with usage of GitHub issues. KS stands for “knowledge seeker”, an KR for “knowledge request”. The process scope concerns just the particular individual who originate the KR thus the process is regarded finished as the person stops participating.

The process aims is timely project knowledge base (PKB) renewal. By PKB I mean an abstract notion unifying for collection of communication artifacts and community vision as well as its members competency. As can be seen from the picture, the fact that searching information on whether the missing piece of knowledge has been request before is optional, never to mention the KS resource is defined willingly by its primary owner. Thus, communication does not always create brand-new knowledge for the entire community. Still, even repeated requests contribute to the identification of the most painful topics, therefore, contribute value creation, for example, through the prioritization of tasks or defining most relevant content for user guide. The metrics describing the process should focus community robustness in terms of knowledge gaps identification.

**Problem level**

For the issue level, I assume that knowledge internalization acts in the form of present issues views and mentions are toppling’s. The issue is an initial knowledge externalization act and thus represents an avalanche that is characterized by its lifetime (open duration) and size measured in the number of comments that correspond to knowledge combination stage are nothing, but subsequent externalization acts provoked by the initial one. The waiting-times are concerned to be elapsed times between two subsequent issues.

**Repository level**

At this point, I adopt the approach used by Tadić et al. (Tadić et al. 2017). Namely, I construct a time series that indicates a number of issues created in each time interval of fixed length (time bins). The threshold-based method with threshold value of 0 was both for events detection and waiting-time measuring implying that the time series segment is concerned to be an avalanche given during corresponding time interval the number of issues per bin does not drop to the zero-level. (Aschwanden 2011) The number of consequent bins with non-zero issues number is thus avalanche duration and the total number of issues created withing the period is avalanche size.

## Model validation

### Process semantics

As it was mentioned before, it is matching process semantics rather than observable statistics that appears to be not only necessary, but also sufficient condition for SOC to be qualified. SOC takes place in open dissipative nonlinear systems. Let me elaborate on each feature one by one.

**OS project as an open system**

Firstly, the system openness stands for external energy supply that gradually drives the system towards the critical state and is commonly referred as loading. Concerning the sandpile, it is toppling that preform the function, and touching on OS project community it can be inflow of people, a as well as the ongoing knowledge inter-/externalization transfer that constitute loading process.

**Dissipation and knowledge loss**

Dissipation implies that there is “energy” loss. In the example with sandpile the system is dissipative due to friction that stops sipping once the slope is bellow critical value. In terms of knowledge development, to my mind two key mechanisms can be named responsible for dissipation. To begin with, it is non-identical information perception due to wide variety of reasons that range from cultural biases and expertise to concurrent mood of interlocutors that result in mutual misunderstanding and information loss. Thus, it impedes knowledge transfer. Another reason is limited attention capacity of individuals. (Fronczak et al. 2005) Human beings can only bear in mind a finite number of most recent concerns; hence they cannot fully comprehend the all the activities going on withing the community and compelled to restrict their scope of interest. For the same reason, I concern the assumption that model should be based on cellular automaton implying nearest neighbor interaction is valid. Therefore, only a fraction of available externalized knowledge is to be internalized, while the remaining part is pushed out of circulation and consequently dissipates. Never to mention that is not only newly introduced triggers within the project, yet it what is happening in real-life that distracts participants as well.

**Non-linearity in knowledge development dynamics**

Nonlinear dynamics is characterized by highly volatile input-output ration. In other words, the systems response is unpredictable and depends on its own internal state and interactions of the constituent parts rather than the input value. The complex interior dynamics leads to instabilities that occasionally results in events of catastrophic size. Just as it happens that the last piece of mental puzzle takes its place. The complexity of co-evolving cognitive, social, and emotional impetus has been studied. (eg. Canbaloğlu et al 2022, Treur 2016, Bosse et al. 2007)

Accordingly, the proposed model is sound with both knowledge development process built on SECI model and definitive features of SOC-susceptible systems. Yet it is statistics that SOC reveals itself by, hence, we should pass to the numerical experiment.

## Experimental evidence

### Methodology

To ensure that the collection of artifacts to be investigated in this study is representative, following three criteria for repository were proposed. The must have users besides its maintainers and be at least 5 years old to make sure that is non-trivial itself and guarantee that there is rich data available. Finally, there should be active communication taking place. The latter I measure in the number of issues that must not be under 10 000. Thereby, Go repository by Golang[[16]](#footnote-17) project was chosen. It was founded in 2008 and is still actively developing. There are over 50k issues in the repository backlog and more that 11k have been closed within a year. The latter constitute the sample I use for my study as due to definition of avalanche duration I propose for issue level the issue framework. Touching on the period longevity, one-year limited was proposed by GitHub API limitations. Still, I suppose that the sample of 11k issues with over 20k comments should be sufficient. In case of project-wide level I choose smaller time bins that Tadić setting 60 minutes per bin versus 100 minutes, the correction is modest, still the time spans investigated are also close enough (1 year versus 15 months), hence the time series must be approximately of the size.

All the data was collected and prepared with Python programming language. The executable notebook authored by the student is available by the link: [*https://github.com/l-o-lenailootolarie/The-Self-organised-Criticality-Perspective-on-Knowledge-Development-and-Governance-in-Open-Source-Pr.git*](https://github.com/l-o-lenailootolarie/The-Self-organised-Criticality-Perspective-on-Knowledge-Development-and-Governance-in-Open-Source-Pr.git)*.* Schematically data processing is depicted on Figure 13.

Diagram

Description automatically generated

Figure 13 - Data collection and processing. The intermittent conversion into excel format was needed due to limitations imposed by API that does not allow for sufficient number of calls to original PyGithub objects.

## Results and discussion

### Chart Description automatically generatedProblem Level

*Figure 14 - Assuming that issue represents an avalanche one obtains following avalanche size (total comments number) and avalanche duration (time lapse between the issue is opened and closed)*

Statistics obtained from empirical data showed that probability function distributions for total number of comments associated with an issue and issues live time representing avalanche size and avalanche duration respectively are powerlaw-like one, thus, consistent with SOC. (see Figure 14) Obviously, the upper and lower cutoffs can be observed, yet that is just natural due to limited system size, and traditionally is attributed to finite size effect.

Touching on the waiting time distribution, the experiment demonstrates that probability mass distribution for elapsed times is best consistent with Weibull distribution. (Figure 15-17) That finds does not SOC, as despite the classical SOC model requiring statistics of waiting times to consistent with Poisson statistics, it has been revealed that powerlaws may be observed in a SOC system with due to nonstationary driver. (e.g., Bartolozzi et al. 2005) In the same time, the concept of “time-to-failure” distribution is even more relevant given the PFPS approach-base perspective. Indeed, an issue indicates presence of a gap between current and desired state, thus resembles failure of concurrent knowledge base sufficiency. In fact, Weibull distribution has been connected to software development process by describing lead times distribution in projects led in Kanban style. (Magennis 2012) The lead time stands for the period from accepting customer commitment to final delivery. Depending on the distribution parameter measurement can made to make conclusions on the project teem maturity and robustness. Meanwhile, the knowledge base failure identification and reporting is a both a sort of feedback and customers’ contribution to value co-creation process. Therefore, the waiting-time Weibull distribution on the community side complements the other one on the maintainers and seem to be candidate for community involvement evaluation.

A picture containing shape

Description automatically generated

Figure 15 - Problem level. Waiting-times distribution

.

The empirically derived distribution is fitted by Weibull distribution with hape parameter and scale parameter . The model has and .

Chart

Description automatically generated

Figure 16 - Fitted Weibull distribution probability density function

Chart, line chart

Description automatically generatedFigure 17 - Weibull line fit.

In case of lead times, David J Anderson, Originator of the Kanban Method, and co-creator of the Kanban Maturity Model claims that corresponds to a longer tail implying lower predictability and greater impact from delay.[[17]](#footnote-18) Therefore, the situation is associated with project methodology immaturity and overall project unhealthiness. Mr. Anderson claims that the parameter values reflect pre-Kanban management approach and can be improved by introduction of certain tools and practices. This makes sense as OS communities are chaotic in its activities and voluntary in terms of involvement in general, moreover the project investigated in the current study has just recently adopted issue templates for user convenience. Thus, one may expect relatively poor community performance. Nevertheless, the parameter value should be compared against benchmark for developing deeper insights.

### Repository Level

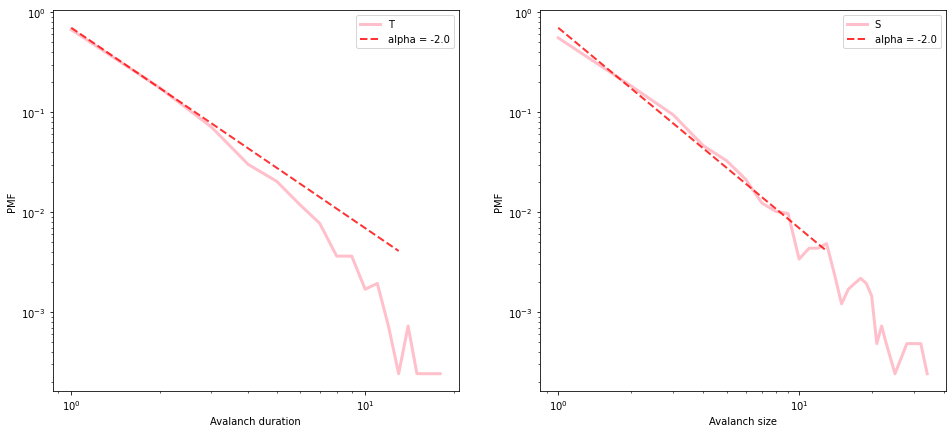
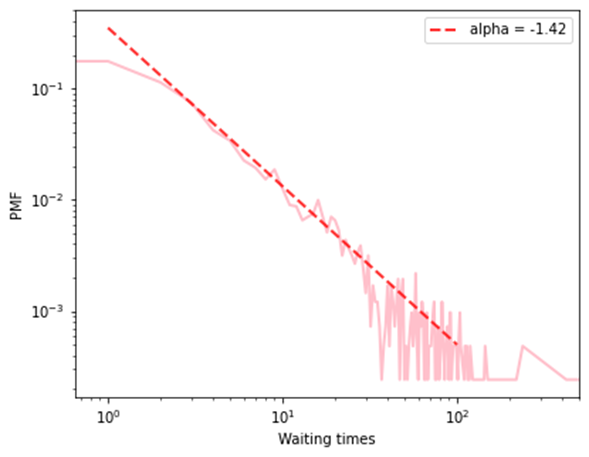
Powerlaw behavior is observed not only on the single issues level but on the project level as well. (Figure 18) The scaling exponent values is about 2 that agrees with theoretical estimation by Fronczak, thus supporting hypothesis on another universality class in SOC-subjected systems. (Fronczak et al. 2006) Moreover, by slope value the shape of observed sample of time intervals suggests consistency with with non-stationary Poisson process, namely that with exponentially varying or -funfcion shaped occurrence rate. (Nurhan et al. 2021) (Figure 19)

Figure 18 - Project level distributions. Given an avalanche is defined as a series of time periods within those the activity level does not drop to zero line, one obtains probability mass functions.

I would expect that the rate would be periodic increasing around release dates, yet the sharpness of rises supports the hypophysis of inherent SOC-subjected process. What I mean is that both on the microscopic issue level and macroscopic project-wide level the community knowledge development process displays SOC phenomena with the lower activity being a driver for the upper one. The inherit complexity might explain the synergy observed in knowledge-ecosystems (Llewellyn & Erkko 2020) and probably lead to self-similarity in community activity patterns (Sowe et al. 2007).

Figure 19 - Repository level. Waiting-times distribution.

****

**Conclusion**

The current study seeks to provide insight in underling mechanism of knowledge developing process in OS communities. In this view it provides detailed description as well as linking the three research domains in a logical manner. The generic framework proposed is then used for setting the experimental part. The experiment was conducted distinguishing between two levels of knowledge communication, namely, project-wide and problem level. In the first case the study showed that observable statistics is consistent with SOC phenomena, moreover the value of obtained scaling exponent is in line with the supposed to be characteristic for a new class of universality in SOC systems which is “in-network activity” class. (Fronczak et al. 2006). The waiting-time distribution, in its turn, suggests SOC phenomena is not bounded to the particular level of analysis rather the system appears to be self-similar in terms of its intrinsic evolution. Therefore, the inherit non-linearity explains the synergy effects that is peculiar to OS projects as knowledge ecosystems. In the second case, the exponents do not meet the values Fronczak et al. Still, the waiting-time distributing that has been shown to be consistent with Weibull distribution is capable to provide valuable tool for assessing community robustness in terms of strategic knowledge development.

**Resources**

1. Andjelković M, Tadić B, Mitrović Dankulov M, Rajković M, Melnik R (2016) Topology of Innovation Spaces in the Knowledge Networks Emerging through Questions-And-Answers. PLoS ONE 11(5): e0154655. <https://doi.org/10.1371/journal.pone.0154655>
2. Aschwanden M. ed. (2011) Self-Organized Criticality in Astrophysics: The Statistics of Nonlinear Processes in the Universe. Springer Praxis Books.
3. Atreyi, K., Bernard C.Y.T. (2004). A review of metrics for knowledge management systems and knowledge management initiatives. In: *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, p. 80238a.
4. Babai, L., & Toumpakari, E. (2010). A Structure Theory of the Sandpile Monoid for Directed Graphs.
5. Barabasi A. L., Albert R. (1999) Emergence of Scaling in Random Networks. *Science*, v. 286, p. 509- 512.
6. Bak, P., Tang, C. and Wiesenfeld, K. (1987) Self-organized criticality - An explanation of 1/f noise. *Physical Review Letters.* 59/27, 381–384
7. Bak, P., Tang, C. and Wiesenfeld, K. (1988) Self-organized criticality. *Physical Review A*. 38/1, 364–374
8. Bangnold, R.A. (2005). The Physics of Blown Sand and Desert Dunes (Dover Earth Science). London: Dover Book.
9. Becerra-Fernandez, I., Leidner, D.E., & Leidner, D. (2008). Knowledge Management: An Evolutionary View (2nd ed.). New York: Routledge.
10. Bianconi G. and Marsili M. (2004) Clogging and self-organized criticality in complex networks. *Physical Review* E 70, 035105(R).
11. Bollobas B. (1998) Modern Graph Theory. New York: Springer.
12. Bosse, T., Memon, Z.A., & Treur, J. (2007). A Two-Level BDI-Agent Model for Theory of Mind and its Use in Social Manipulation. Proceedings of the AISB 2007 Workshop Mindful Environments, pp. 335-342.
13. Camazine, S., Deneubourg J.-L., Franks N. R., Sneyd J., Theraula G., E. Bonabeau. (2001) Self-Organization in Biological Systems (Princeton Studies in Complexity, 38)
14. Canbaloğlu, G., Treur, J., & Roelofsma, P.H. (2022). Computational modeling of organisational learning by self-modeling networks. Cognitive Systems Research, 73, 51-64.
15. Carpendale J.I.M. and Müller U., editors. (2013) Social Interactions and the Development of Knowledge. New Jersey: Lawrence Erlbaum Associates, Inc. Mahwah
16. Cepeda, G., & Vera, D. (2007). Dynamic capabilities and operational capabilities: A knowledge management perspective. *Journal of business research*, 60(5), 426-437. doi:10.1016/j.jbusres.2007.01.013
17. Chang, T. (1992) Low-dimensional behavior and symmetry breaking of stochastic systems near criticality - can these effects be observed in space and in the laboratory? *IEEE Trans. on Plasma Science* 20, 691
18. Cooperrider, D. & Srivastva, S. (1987). Appreciative Inquiry in Organizational Life. Research in organization change and development. Research in Organizational Change and Development, Vol.1, pages 129-169.
19. Dankulov, M., Melnik, R. & Tadić, B. (2015) The dynamics of meaningful social interactions and the emergence of collective knowledge. *Scientific Reports* 5,12197. https://doi.org/10.1038/srep12197
20. Davenport, T.H. and Grover, V. (2001) General Perspectives on Knowledge Management: Fostering a Research Agenda. Journal of Management Information Systems, 18, 5-23.
21. Dhar D. (1990) Self-organized critical state of sandpile automaton models. *Physical Review Letters*, 64(14):1613–1616,
22. Ding W., Liang P., Tang A., van Vliet H., and Shahin M. (2014) How do open source communities document software architecture: An exploratory survey, in: *Proceedings of the 19th International Conference on Engineering of Complex Computer Systems (ICECCS*), Tianjin, China, IEEE, pp. 136-145.
23. Dmitriev, A., Kornilov, V., Dmitriev, V., and Abbas, N. (2022) Early Warning Signals for Critical Transitions in Sandpile Cellular Automata. *Frontiers in Physics*, vol. 10, No. 839383. Article 839383. doi:10.3389/fphy.2022.839383.
24. Dorogovtsev S.N. (2010) Lectures on Complex Networks. *Oxford Master Series in Statistical, Computational, and Theoretical Physics*, Oxford Univ. Press.
25. Feigenbaum, J.A. (2003). Financial Physics. Reports on Progress in Physics, 66, 1611-1649.
26. Fitzerald, B. (2004). A critical look at open source. *Computer* 37 (7), 92–94.
27. Fronczak, P., Fronczak, A., & Hołyst, J.A. (2006). Interplay between network structure and self-organized criticality. *Physics Review* E 73, 046117
28. Foss, N. J. (2007). The emerging knowledge governance approach: Challenges and characteristics. *Organization*, 14(1), 29-52. doi:10.1177/1350508407071859
29. Foss, N. J., & Michailova, S. (2009). Knowledge governance: Processes and perspectives. Oxford; New York: Oxford University Press.
30. Foster, L. & Wiewiora, A. & Donnet, T. & Chang, A. (2018). Exploratory Investigation of Innovation Management in a Project-Based Organisation. 10.5204/thesis.eprints.122923.
31. Grandori, A. (2001). Neither hierarchy nor identity: Knowledge-governance mechanisms and the theory of the firm. *Journal of Management and Governance*, 5(3), 381-399. doi:10.1023/A:1014055213456
32. Grandori, A. (2009). Poliarchic governance and the growth of knowledge. In N. J. Foss & S. Michailova (Eds.), Knowledge governance: Processes and perspectives. Oxford: Oxford University Press.
33. Grant R. M. (1996) Toward a Knowledge-Based Theory of the Firm. *Strategic Management Journal*. Vol. 17, Special Issue: Knowledge and the Firm, pp. 109-122.
34. Healy, K., Schussman, A. (2003). The ecology of open-source software development. (http://www.kieranhealy.org/files/drafts/oss- activity.pdf, - дата обращения 28.04.2022)
35. Jensen C., King S., and Kuechler V. (2011) Joining free/open source software communities: An analysis of newbies' first interactions on project mailing lists, in: *Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS*), Kauai, USA, IEEE, pp. 1-10.
36. Jin, Z., T.P. Charlock, K. Rutledge, G. Cota, R. Kahn, J. Redemann, T. Zhang, D.A. Rutan, and F. Rose (2005) Radiative transfer modeling for the CLAMS experiment. *Journal of Atmosphere Science*, 62, no. 4, 1053-1071
37. Kalinin N, Guzmán-Sáenz A, Prieto Y, Shkolnikov M, Kalinina V, Lupercio E. (2018) Self-organized criticality and pattern emergence through the lens of tropical geometry. Proceedings of National Academy of Science U S A. Aug 28;115(35):E8135-E8142. doi: 10.1073/pnas.1805847115. Epub 2018 Aug 15. PMID: 30111541; PMCID: PMC6126730.
38. Katz, J. I. (1986) *Journal of Geophysical Research: Solid Earth*, 91(B10), 10412.
39. Kim, E. (2003). An introduction to open-source communities. Blue Oxen Associates. ( [http://www.blueoxen.org/research/00007/BOA- 00007.pdf](http://www.blueoxen.org/research/00007/BOA-%2000007.pdf) – дата обращения 28.04.2022)
40. Kimmerle J., Kress U., and Held Ch. (2010) The interplay between individual and collective knowledge: technologies for organisational learning, *Knowledge Management Research & Practice*. 8:33-44
41. Koch S. (2004) Agile principles and open source software development: a theoretical and empirical discussion. *Extreme Programming and Agile Processes in Software Engineering. Proceedings of the 5th international conference XP (2004),* 85-93. Berlin: Springer-Verlag (LNCS 3092)
42. Kozlov, V., Kuznetsov, N. & Motygin, O. (2004) On the two-dimensional sloshing problem. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 460, 2587–2603
43. Krogh, G., Spaeth, S., Lakhani, K. (2003). Community, joining, and specialization in open source software innovation: a case study. *Research Policy* 32, 1217–1241.
44. Lakhani, K., Hippel, E. (2003). How open source software works: ‘‘free’’ user-to-user assistance. *Research Policy* 32, 923–943.
45. Lambiottea, R. & Panzarasab, P. (2009) Communities, Knowledge Creation, and Information Diffusion. *Journal of Informetrics*, Volume 3, Issue 3, July 2009, Pp. 180-190
46. Lefebvre, R., & Mukamel, S. (1987). Stochasticity and intramolecular redistribution of energy. Nato Science Series. Springer.
47. Llewellyn T., & Erkko A. (2020). Innovation ecosystems in management: An organizing typology. 10.1093/acrefore/9780190224851.013.203.
48. Lui, A. T. Y., Chapman S. C., Liou K., Newell P. T., Meng C. I., Brittnacher M., and Parks G. D. (2000) Is the dynamic magnetosphere an avalanching system? *Geophysics Research Letters*, 27, 911–914.
49. Lyotard, J.-F. (1984). The Postmodern Condition: A Report on Knowledge. University of Minnesota Press.
50. Madey G., Freeh V., and Tynan R. (2004) Modeling the F/OSS Community: A Quantitative Investigation. In *Free/Open Source Software Development*, ed., Stephan Koch. Idea Publishing.
51. Magennis, T. (2012). Managing Software Development Risk using Modeling and Monte Carlo Simulation. Lean Software and Systems Conference. Boston, MA, USA. Conference Paper. ([https://www.researchgate.net/publication/270687052\_Managing\_Software\_Development\_Risk\_using\_Modeling\_and\_Monte\_Carlo\_Simulation - дата обращения 5.05.2022](https://www.researchgate.net/publication/270687052_Managing_Software_Development_Risk_using_Modeling_and_Monte_Carlo_Simulation%20-%20дата%20обращения%205.05.2022))
52. Man, A.P., Berends, J.J., Lammers, I., Raaij, E.M., & Weele, A.J. (2008). Knowledge and innovation in networks : a conceptual framework. In Knowledge management and innovation in networks. Cheltenham: Edward Elgar Publishing.
53. McGrath, K. (2005). Doing critical research in information systems: a case of theory and practice not informing each other. *Information System Journal*, Vol.15, 2, 85-101.
54. Newman M. E. J., Girvan M. (2002) Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* (*PNAS)*, 99 (12) 7821-7826.
55. Nickerson, J. & Zenger, T. & Heiman, B. (2009). Governing Knowledge Creation: A Problem‐Finding and Problem‐Solving Perspective. In Knowledge Governance. pp.25-46. 10.1093/acprof:oso/9780199235926.003.0002.
56. Nichols, D., Twidale, M. (2003). The usability of open source software. Firstmonday 8 (1).
57. Nonaka, I., & Konno, N. (1998). The Concept of “Ba”: Building a Foundation for Knowledge Creation. *California Management Review, 40*, 40 - 54.
58. Nurhan, Y.I., Johnson, J.R., Homan, J.R., Wing, S., & Aschwanden, M.J. (2021). Role of the Solar Minimum in the Waiting Time Distribution Throughout the Heliosphere. *Geophysical Research Letters*, 48, 16.
59. Paczuski M. and Hughes D. (2004) A heavenly example of scale-free networks and self-organized criticality. *Physica A* 342, 158-163.
60. Pemsel, S., & Müller, R. (2012). The governance of knowledge in project-based organizations. International journal of project management, 30(8), 865-876. doi:10.1016/j.ijproman.2012.02.002
61. Rashid, M., Clarke, P.M., & OConnor, R. (2017). Reducing Knowledge Loss in Open Source Software. (<https://www.semanticscholar.org/paper/Reducing-Knowledge-Loss-in-Open-Source-Software-Rashid-Clarke/8509c22efa839b81bb8c74771ace1b5c598c8f54> - дата обращения 20.04.2022)
62. Schmitz, C. (2005). Towards Self-Organizing Communities in Peer-to-Peer Knowledge Management. (<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.783&rep=rep1&type=pdf> – дата обращения 25.04.2022)
63. Sowe, S.K., Stamelos, I., & Angelis, L. (2008). Understanding knowledge sharing activities in free/open source software projects: An empirical study. *Journal of Systems and Software*, 81, 431-446.
64. [Sunindyo](https://dblp.uni-trier.de/pid/05/7096.html), W.D., [Moser](https://dblp.uni-trier.de/pid/72/5603.html), T., [Winkler](https://dblp.uni-trier.de/pid/60/2067.html), D., [Mordinyi](https://dblp.uni-trier.de/pid/08/1107.html), R. (2013) Project Progress and Risk Monitoring in Automation Systems Engineering. [*SWQD,*](https://dblp.uni-trier.de/db/conf/swqd/swqd2013.html#SunindyoMWM13)30-54.
65. Šuvakov, M., Mitrovic ́, M., Gligorijevic ́, V. and Tadic B. (2012) How the online social networks are used: dialogues-based structure of Myspace. *Journal of the Royal Society Interface*, 10(79):20120819,
66. Swan, J., Newell, S., Scarbrough, H., & Hislop, D. (1999). Knowledge management and innovation: networks and networking. *Journal of Knowledge Manageme*nt*, 3*, 262-275.
67. Tadić, B. (2019) Self-organised criticality and emergent hyperbolic networks: blueprint for complexity in social dynamics. *European Journal of Physics*, vol. 40, no. 2, p. 024002. doi:10.1088/1361-6404/aaf144.
68. Tadić, B., Dankulov, M.M., and Melnik R. (2017) Mechanisms of self-organized criticality in social processes of knowledge creation. *Physical Review*, E**96**, 032307
69. Thatchenkery, T. (2005). Appreciative Sharing of Knowledge: Leveraging Knowledge Management for Strategic Change. Taos Institute.
70. Thatchenkery, T., & Chowdhry, D. (2007). Appreciative Inquiry and Knowledge Management. Cheltenham: Edward Elgar Publisher.
71. Treur, J. (2020). Modeling higher order adaptivity of a network by multilevel network reification. *Network Science*, 8, S110 - S144.
72. Treur, J. (2016). Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Series on Understanding Complex Systems. Springer Publishers.
73. Watkins R. (2015) What is Open Source Software and is it usable? UPA Voice, 5 (1). (<http://www.upassoc.org/upa_publications/upa_voice/volumes/5/issue_1/open_source.htm> - дата обращения 15.04.2022)
74. Xu, J., & Madey, G.R. (2004). Exploration of the Open Source Software Community. *NAACSOS Conference 2004* (<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=84DB2EADAEE05C4F529FAD0E41102D3E?doi=10.1.1.72.5051&rep=rep1&type=pdf> – дата обращения 20.04.2022)
75. Yoo, Y., Boland, R. J., Lyytinen, K., & Majchrzak, A. (2012). Organizing for innovation in the digitized world. *Organization Science*, 23(5): 1398-1408.
76. Yutaka, Y., Makoto, Y., Takeshi, S., and Toru, I. (2000) Collaboration with Lean Media: how open-source software succeeds," in: *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, ACM, Philadelphia, Pennsylvania, United States.
77. Zeldin, T. (1999) How work can be made less frustrating and conversation less boring. British Medical Journal (BMJ). 18; 319(7225): 1633–1635.

1. https://www.bcg.com/publications/2021/open-source-software-strategy-benefits [↑](#footnote-ref-2)
2. Ibid., 3 [↑](#footnote-ref-3)
3. https://base.garant.ru/6746035/ [↑](#footnote-ref-4)
4. https://digital.gov.ru/ru/documents/4548/ [↑](#footnote-ref-5)
5. Проект\_стратегии%2C\_15\_сентября.pdf [↑](#footnote-ref-6)
6. https://russiaos.ru/summit/translation/ [↑](#footnote-ref-7)
7. https://habr.com/ru/company/sberbank/blog/418777/ [↑](#footnote-ref-8)
8. https://yandex.ru/dev/catboost/ [↑](#footnote-ref-9)
9. https://youtu.be/1mPxNc3cNZE [↑](#footnote-ref-10)
10. https://www.youtube.com/watch?v=H0R7QgfoXaU [↑](#footnote-ref-11)
11. https://www.youtube.com/watch?v=mMfG38wj8Dg&t=4790s [↑](#footnote-ref-12)
12. https://www.tadviser.ru/index.php/Компания:SourceForge#cite\_note-b-1 [↑](#footnote-ref-13)
13. https://bitbucket.org/blog/celebrating-10-million-bitbucket-cloud-registered-users [↑](#footnote-ref-14)
14. https://jelvix.com/blog/bitbucket-vs-github-vs-gitlab [↑](#footnote-ref-15)
15. https://github.wiki/en/enterprise-cloud@latest/get-started/quickstart/communicating-on-github [↑](#footnote-ref-16)
16. https://github.com/golang/go [↑](#footnote-ref-17)
17. https://djaa.com/what-we-know-about-duration-workflows/ [↑](#footnote-ref-18)