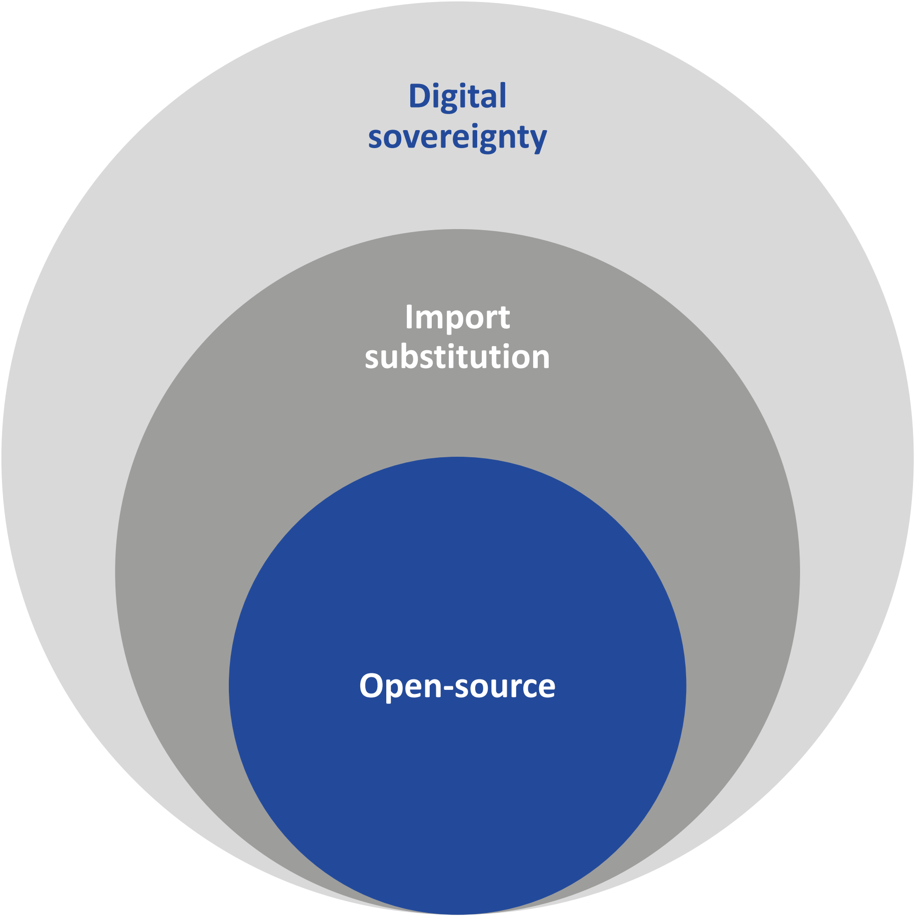
Высшая
школа
бизнеса

High School of Business,
Business Informatics

Moscow
2022

The Self-organised Criticality Perspective on Knowledge Development and Governance in Open-Source Projects

Student: Ekaterina Drobyshevskaya, BBI182
Mentor: Dmitriev Andrey V., Professor



Background

For over 10 years open-source has been formally recognized to have a great potential in terms of development digital sovereignty.

The goal –

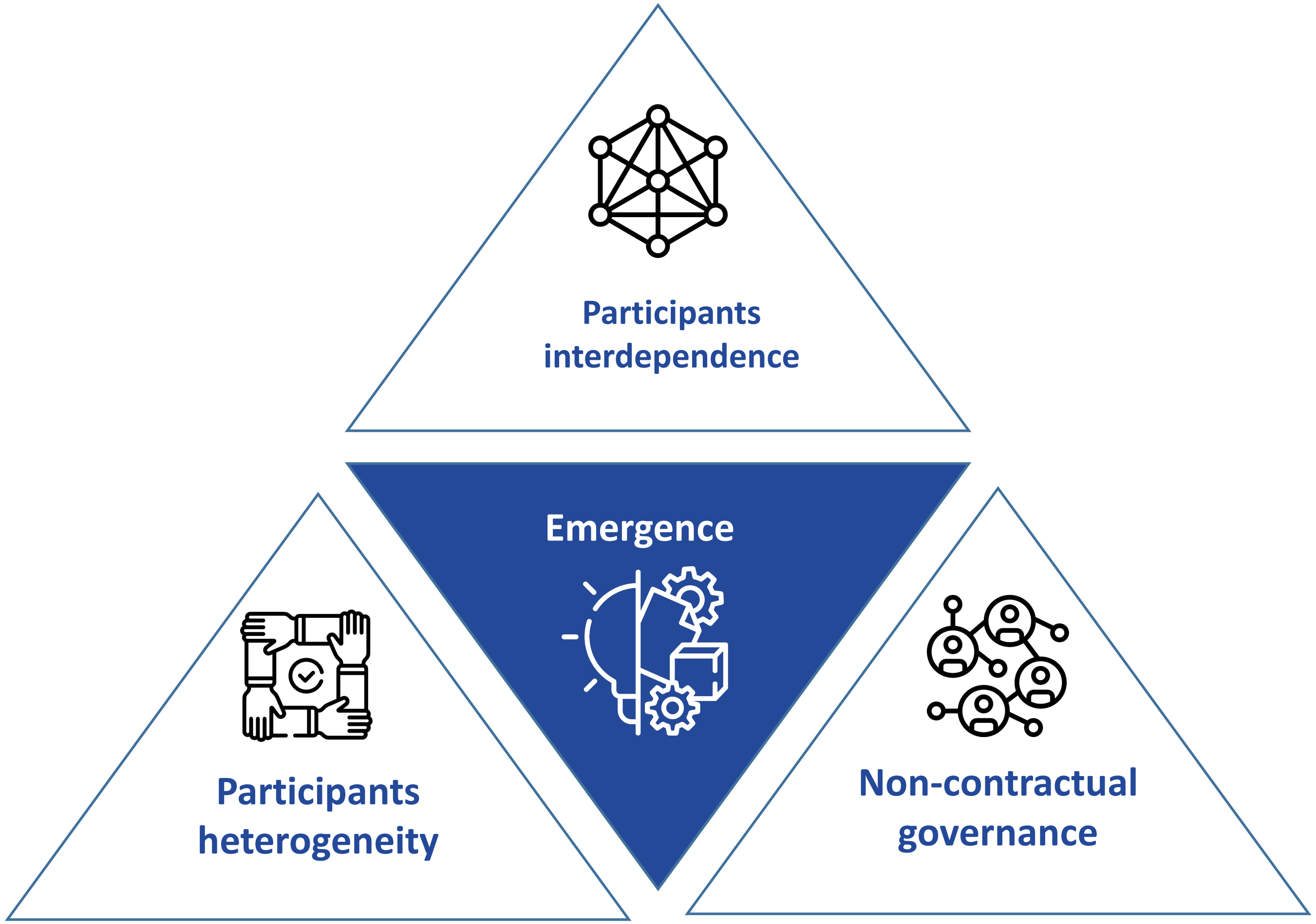
Develop an interpretable model of knowledge development dynamics in OS community and suggest mathematical characteristics appearing to be sound performance metrics.

The tasks set

1. Review common concepts in 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. Test whether associated statistics is consistent with SOC.
5. Interpret obtained results.

The object – Open-Source Community

OS community can be recon as knowledge ecosystem based on both qualitive (Llewellyn & Erkko 2020) and quantitive evidence (Sowe et al. 2007).



The subject – Knowledge governance

Knowledge governance explains the behaviors, rules, and mechanisms of organizations associated with developing, coordinating, and using knowledge within an organization.

(Foss 2007)

According to **Problem Finding Problem Solving Perspective (PFPS)**, problem is opportunities for value creation, thus potential for deliberate value creation lies in finding and solving problems. (Nickerson and Zenger 2004)

PF

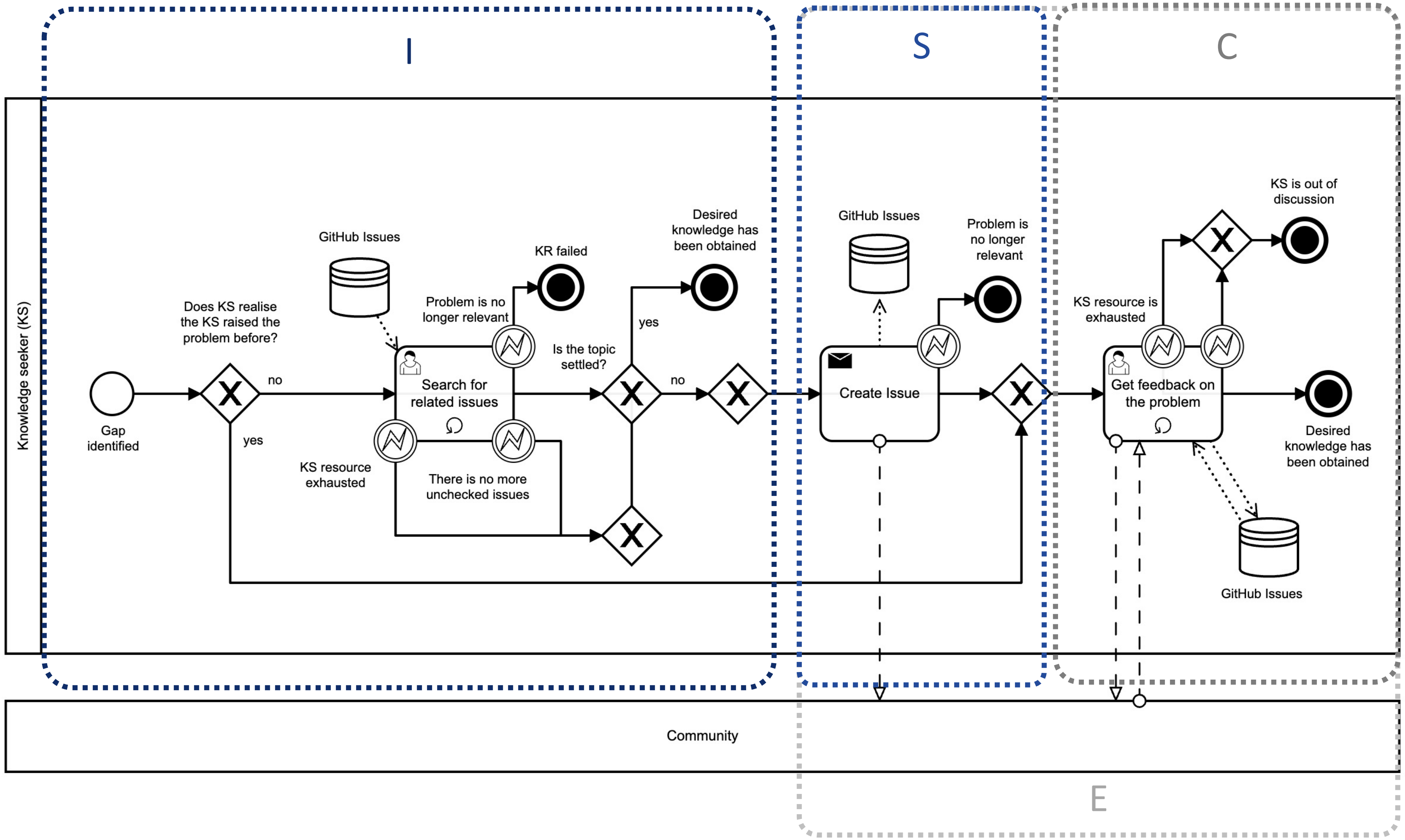
Problem finding is concerned describe in broad terms the knowledge, which has potential for value creation.

The key question:
How to organize a search to identify and select the most relevant problems?

PS

Problem solving deals with knowledge recombination given organizational and case-related peculiarities.

The key question:
How to organize an efficient search for solutions to the problems?



Knowledge development with usage of GitHub issues.

The process & data Collection

The aim of the process is timely identification of insights.

For empirical data GitHub-based repository Go by Golang project has been chosen for being non-trivial and GitHub issues is an activity to analyse due to incorporating strategic significance and robust semantic coverage.

Knowledge development in OS community

Key characteristics

Openness

The inflow of people, a as well as the ongoing knowledge inter-/externalization transfer that constitute loading process.

Dissipation

Non-identical information perception and limited attention capacity of individuals result in mutual misunderstanding and information loss.

Non-linearity

The last piece of mental puzzle takes its place according to the complex interior dynamics emerging from cognitive, social, and emotional impetus (eg. Treur 2016)

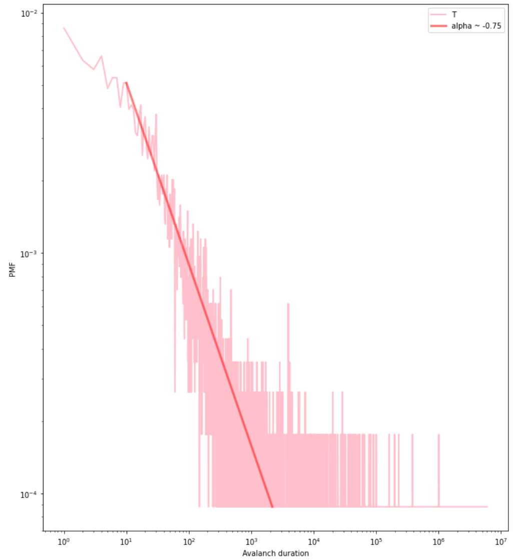
Self-organized criticality in knowledge development

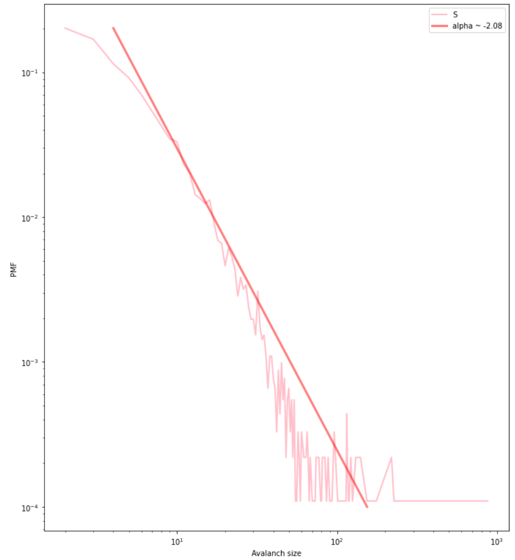
Fronczak et al. suggest that there is another SOC universality class covering “in-network” with theoretically derived scaling exponent value $\alpha = 2$.

<table>
<tr><th>Sandpile model (Bak et al.)</th><th>Tadić et al. 2017</th><th>Project-level model</th><th>Fronczak et al. 2006</th><th>Problem-level model</th></tr>
<tr><td>Sandpile</td><td>Q&A site</td><td>OS community</td><td>Academic community</td><td>OS community</td></tr>
<tr><td>Sand grain</td><td>Agent action</td><td>Issue</td><td>Quantum of idea</td><td>Knowledge internalization act</td></tr>
<tr><td>Avalanche</td><td colspan="2">The period with number of actions taken/issues created is above zero.</td><td>Research area</td><td>Issue</td></tr>
<tr><td>Avalanche size</td><td colspan="2">Total number of issues created/agent actions within the period</td><td colspan="2">Number of associated papers /comments</td></tr>
<tr><td>Avalanche duration</td><td colspan="2">The period duration</td><td>Not stated</td><td>Issue life-time</td></tr>
</table>

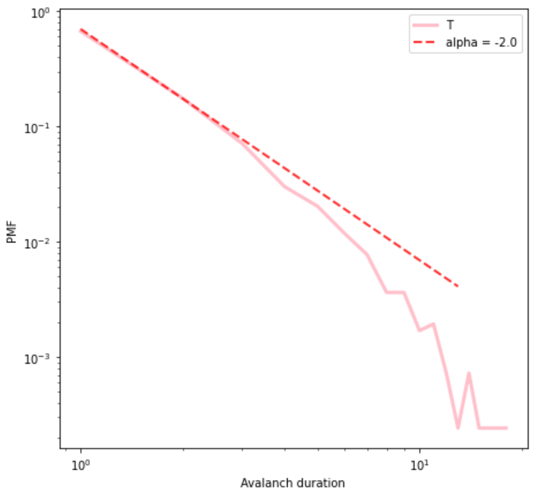
Avalanche size and avalanche duration distributions

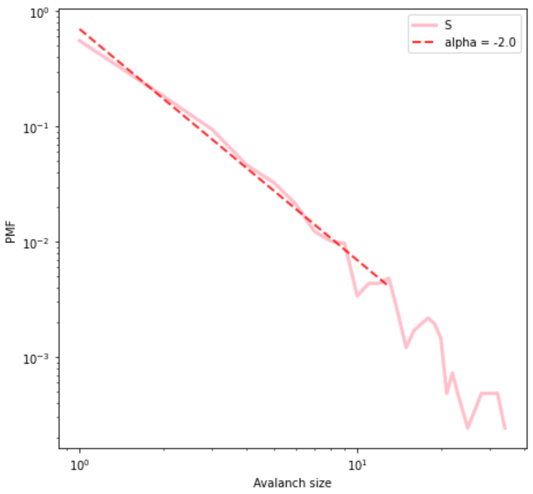
Problem level





Project level





Hypothesis:

The distributions in question obey powerlaw.

Outcome:

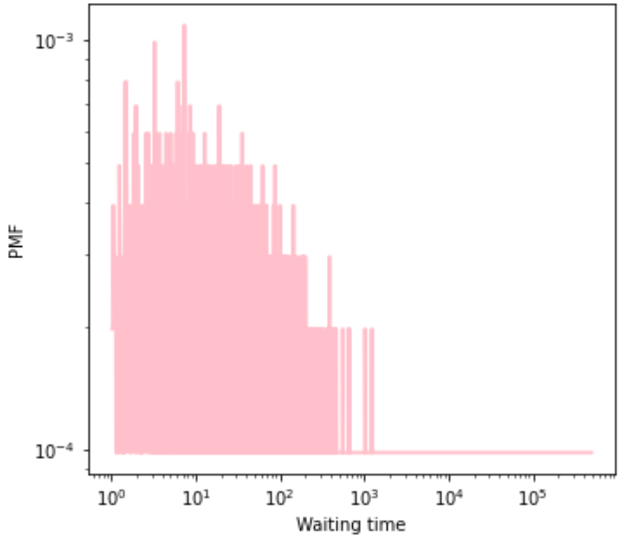
Both avalanche size and avalanche duration* distributions scaling exponent is consistent with that suggested by Fronczak et al. 2006

Waiting-times distribution

Problem level

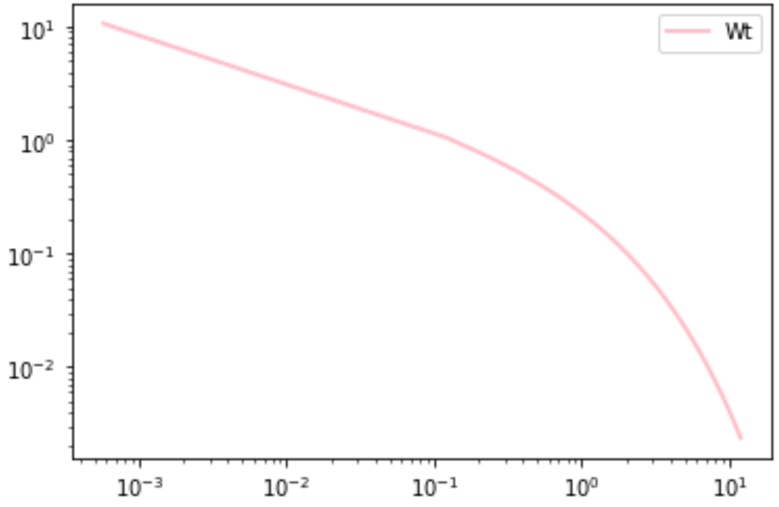
The experiment demonstrates that probability mass distribution for elapsed times is best consistent with Weibull distribution with shape parametr $\beta < 1$.

According to Kanban method, the value of β correspond to low maturity level, yet due to domain specificity benchmark should be estimated before making conclusion.



Probability mass function for problem level waiting-times distribution. (above)

Probability density function for fitted Weibull distribution. (down)

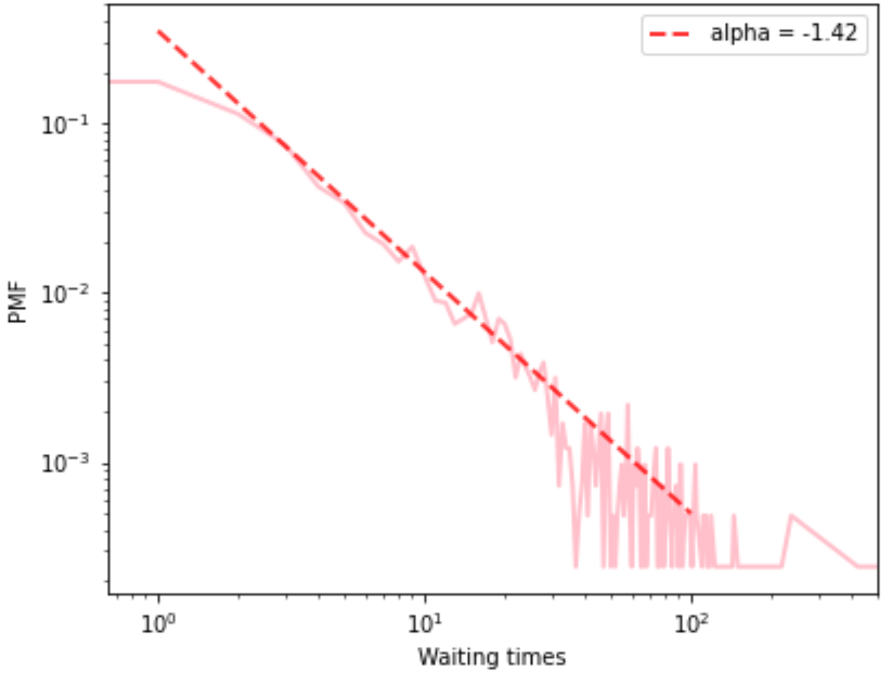


Waiting-time distribution

Project level

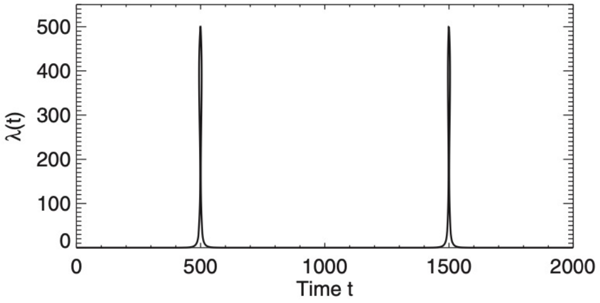
The experiment showed that waiting-times distribution probability mass obeys powerlaw with scaling exponent $\alpha < 1.9$.

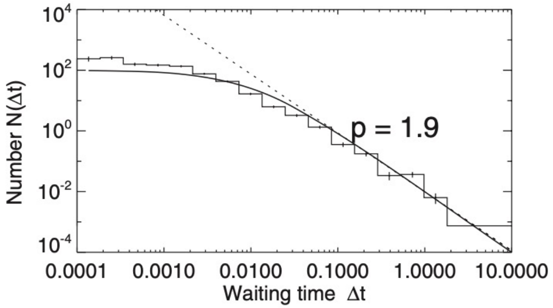
The value suggests non-stationary toppling events occurrence rate and correspond to δ-function like variation and gives ground to suppose the process is inheritedly subjected to SOC.



Project level Waiting-times distribution probability mass function. (on the left)

Poisson processes with δ-function like variation of the occurrence rate λ(t). (Aschwanden 2011, below)





Conclusion

Knowledge development in OS community exhibit definitive system features associated with SOC.

Observable statistics associated with knowledge development in OS community is consistent with SOC phenomena.

Project level waiting-times scaling exponent suggest that the system is self-similar in terms of evolution.

Knowledge development in OS community can be approached from PFPS perspective. Herewith, by being highly involved during PF stage, community contribute to value co-creation.

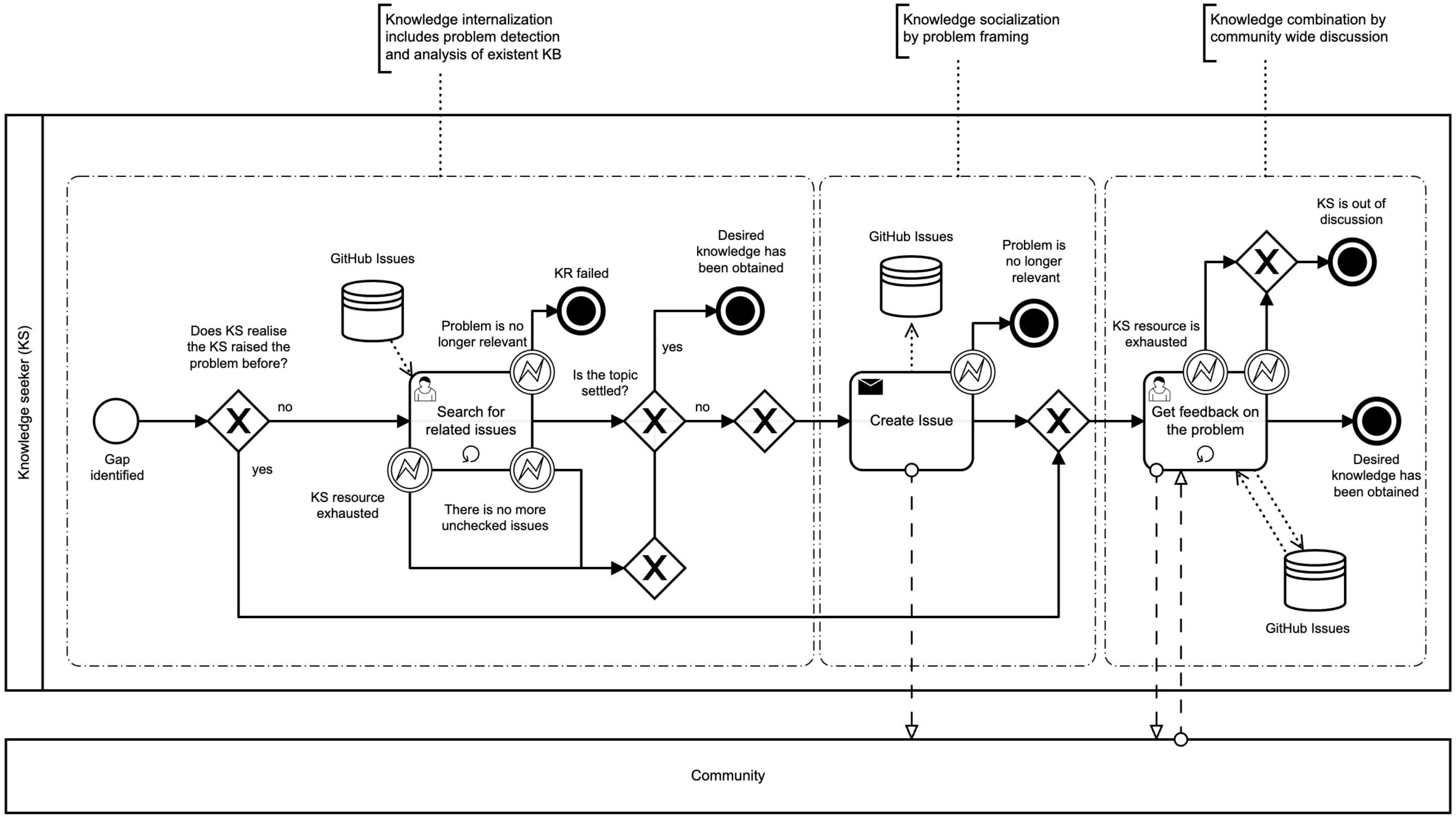
Problem level waiting-times distribution is consistent with Weibull distribution that allows for developing complementary metric for assessing community robustness in terms of identifying insights.

Resources

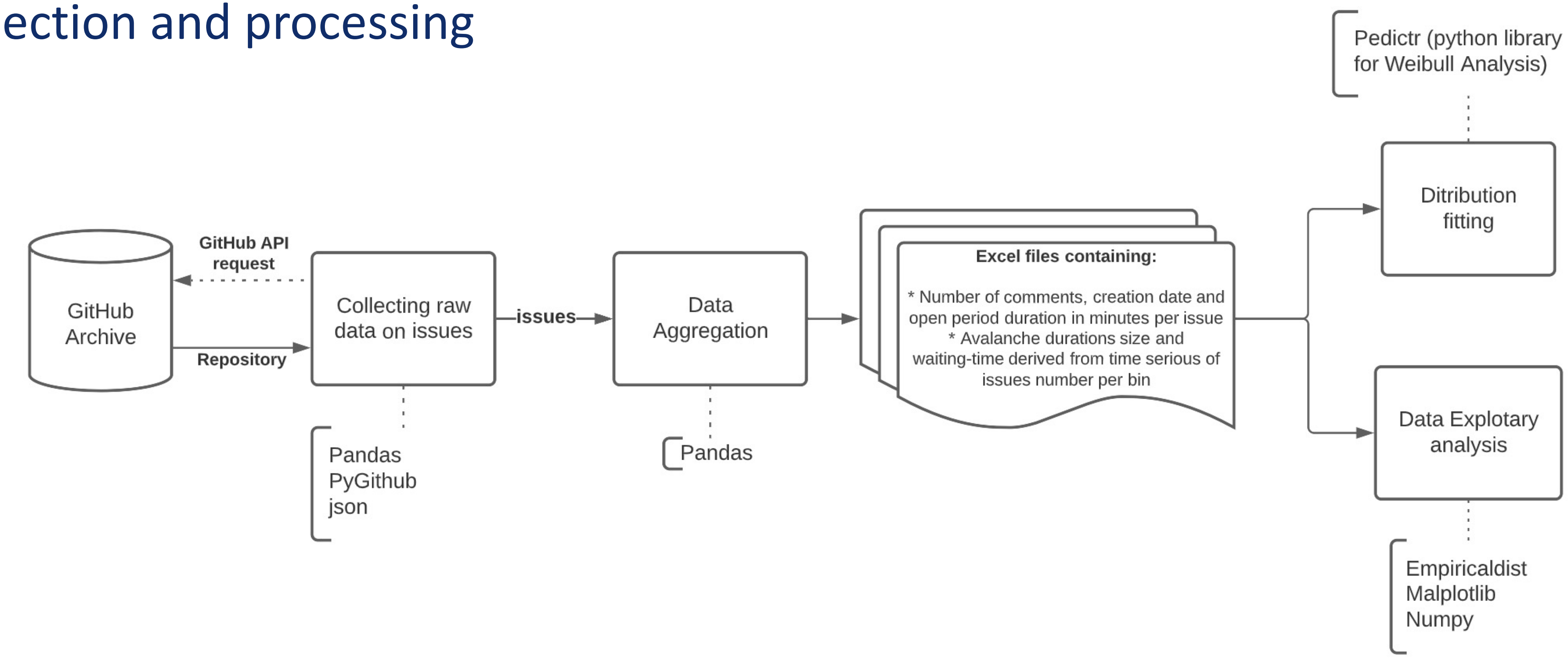
- Aschwanden M. ed. (2011) Self-Organized Criticality in Astrophysics: The Statistics of Nonlinear Processes in the Universe. Springer Praxis Books.
- Foss, N. J. (2007). The emerging knowledge governance approach: Challenges and characteristics. Organization, 14(1), 29-52. doi:10.1177/1350508407071859
- Fronczak, P., Fronczak, A., & Hołyst, J.A. (2006). Interplay between network structure and self-organized criticality. Physics Review E 73, 046117
- Llewellyn T., Erkko A. (2020). Innovation ecosystems in management: An organizing typology. In Oxford Research Encyclopedia of Business and Management. Oxford University Press.
- 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.
- Nonaka, I., & Konno, N. (1998). The Concept of “Ba”: Building a Foundation for Knowledge Creation. California Management Review, 40, 40 - 54.
- 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.
- Tadić, B., Dankulov, M.M., and Melnik R. (2017) Mechanisms of self-organized criticality in social processes of knowledge creation. Physical Review, E96, 032307
- Treur, J. (2016). Network-Oriented Modeling: Addressing Complexity of Cognitive, Affective and Social Interactions. Series on Understanding Complex Systems. Springer Publishers.

Appendix 1. 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.



Appendix 2.
Data collection and processing



https://github.com/l-o-lenailootolarie/The-Self-organised-Criticality-Perspective-on-Knowledge-Development-and-Governance-in-Open-Source-Pr.git

Высшая
школа
бизнеса

Appendix 3.
Sandpile model

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.

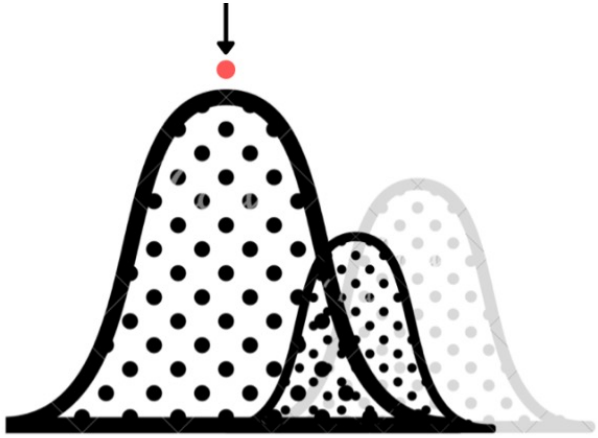


Figure 1 - Step one. Random grain input.

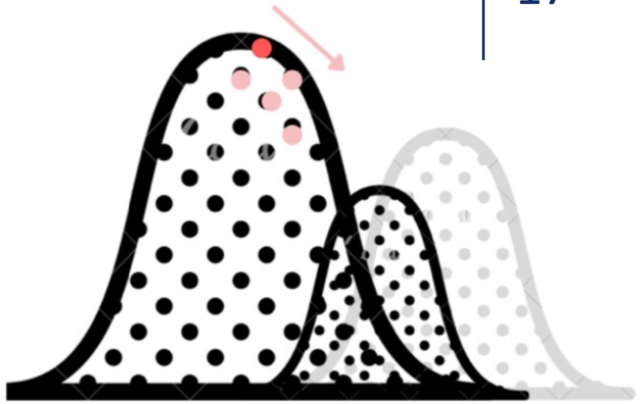


Figure 2 - Step 2. The input drives first "pink" sapping.

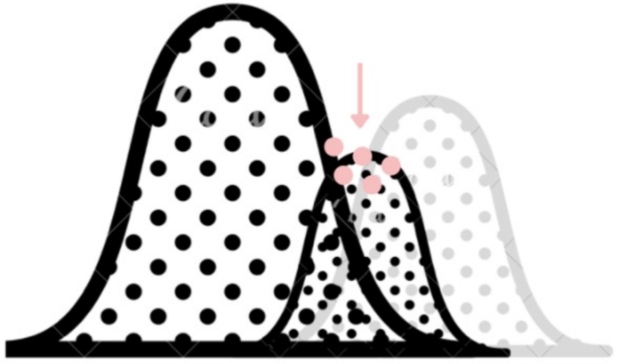


Figure 3 - The pink sapping grains appear to be an input for the second sub-sandpile.

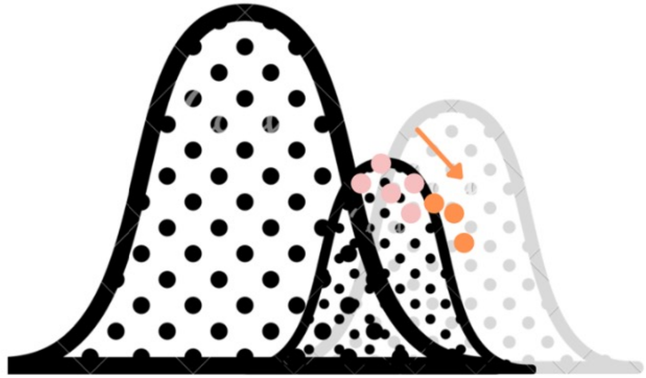


Figure 4 - The pink input leads to critical slope for the second sub-sandpile and "orange" sappings starts.

Appendix 4.
Cellular Automaton

The Cellular automaton model includes following elements:

- S-dimensional regular lattice grid.
- Set $\{z\}_{n^s}$ of «grains» number corresponding to each lattice node.
- z_{crit} – value for critical threshold, or a maximum number of grains that a node can hold.
- Random input in space and time $\Delta z_{i_1..i_s}$

z_{11}	z_{12}	z_{13}
z_{21}	z_{22}	z_{23}
z_{31}	z_{32}	z_{33}

z_{11}	z_{12}	z_{13}
z_{21}	$z_{22}^{*}+1$	z_{23}
z_{31}	z_{32}	z_{33}

$$z_{22}^{*} + 1 = z_{22}^{crit}$$

z_{11}	$z_{12}+1$	z_{13}
$z_{21}+1$	$z_{22}^{*}-4$	$z_{23}+1$
z_{31}	$z_{32}+1$	z_{33}

Appendix 5.
Fitting Weibull distribution

Python library **predictr** used. Total number of avalanches is 10K+.

$$r^2 = 0.954$$

```
from predictr import Analysis
# Data from testing: avalanch elapsed times
failures = diffs_df[0]
# Weibull Analysis
x = Analysis(df=failures, show=True)
x.mrr()
```