

# Social Networks during Software Ecosystems' Death

1<sup>st</sup> Pedro Arantes

FACOM/UFMS

Campo Grande-MS, Brazil  
pedro.arantes@ufms.br

2<sup>nd</sup> Felipe Soupinski

FACOM/UFMS

Campo Grande-MS, Brazil  
felipe\_s@ufms.br

3<sup>rd</sup> Awdren Fontão

FACOM/UFMS

Campo Grande-MS, Brazil  
awdren.fontao@ufms.br

**Abstract**—Software Ecosystems (SECO) depend on platforms that serve as environments for developers' interaction. The SECO may die when the organization that owns the platform does not support the synergy between organizational goals and developers' expectations. The death results in the suspension of vital activities, such as code development and maintenance, impacting developers who lose work, learning, and experience gained. On the other hand, the responsible corporation loses resources invested in SECO. These signs indicate a SECO death is an important event for the community and should be analyzed. This paper reports a GitHub (GH) study focusing on three web SECOS: AngularJS, PhantomJS, and MomentJS. We analyze metrics based on developer community engagement and collaboration to understand what happens in these SECOS before, during, and after the platform's death. From the search questions, we found some directions: Users' recruitment and permanence: Without management and engagement, communities get out of SECOS a few years after your entry, so methods to control and organize the community are needed. Community relationships: The community needs engagement to strengthen and maintain your relationships. Some types of programs are useful methods to engage and encourage users to collaborate between them.

**Index Terms**—software ecosystem, death platform, death ecosystem.

## I. INTRODUCTION

The software ecosystem's platform is an environment that allows users to develop and maintain software projects. Its relevance and continuous evolution are key to attracting the development community to contribute to the project, either through the possibility of public recognition and fame or by evolving soft and hard skills.

If the responsible organization does not apply sustainable strategies to engage and support the synergy between community and organizational objectives, the stimulus may drop and cause ecosystem evasion. Without a workforce, users and other platforms that depend on the software ecosystem need to adapt their work if the static system does not support new requirements.

As a social and cooperative tool, these platforms are useful environments to exchange resources and information between your members once the community works together to create and maintain artifacts around the software ecosystems.

The current literature focuses on mortal behavior, Windows Phone death on Stack Overflow analyses, development metrics, and recycling resources. To improve the state of the art, the main study objective is to investigate social networks

around SECO and understand this phenomenon in the software engineering landscape. The selected SECOS are directed to web development tools.

To conduct this paper, we defined three research questions to answer:

- Can SECO maintain your contributors' activity before, during, and after your death? In the first years of their lives, the projects attract a large number of developers, but they cannot keep their active community for a long time, especially after their deaths.
- Can SECO maintain your contributors' relationship level before, during, and after your death? Users normally connect with only another developer, and the community level decreases after the project's death.
- Do SECO developers build strong relationships? Developers can construct strong relationships with other users, but do not maintain their connections for a long time, resulting in a constant decrease in strength level after a few years of project life.

## II. SECO DEATH

A software ecosystem (SECO) is the environment around a software platform where actors interact with other members and use artifacts. Some of the tools that comprise the SECO make it easier for the community to work together to use, develop, and take care of the software platform.

Stack Overflow (SO) and GitHub (GH) are two examples of tools that encourage collaboration and belong to SECO. SO is a question-and-answer environment around technical questions from developer communities. GH is a development environment that allows users to code and create artifacts, manage changes, control artifact versions, and integrate/implant projects. These software tools include metrics for tracking project activity, improving understanding of SECO health, and community engagement.

We can detect the death of the software if the community stops your activity on these platforms. In GH, "project death" occurs when the community stops development and maintenance for one year (i). On the other hand, the organization that owns SECOS can end it by putting out a public notice (ii).

### III. SOCIAL NETWORKS

A social network is a group of actors and their interactions in a common environment. This concept helps researchers and users understand how a group of people socialize in a determined context, such as social media, work relationships, and other scenarios.

We are going to use the graph concept, a subarea of information sciences, to examine how developers interact with the community and their ability to form connections to collaborate. The social networks will be shown as a group of "nodes," which are users, and "edges," which are the connections between them. The direction of interactions to build relationships and the force in relationships are two other ideas that are talked about.

Using this group of concepts, the movements of developers' immigration and emigration can be tracked and viewed. This factor will enable us to confirm the SECO's health within a determined period.

### IV. RELATED PAPERS

Evertse et al. [3] investigate the SECO principles, practices, and behaviors that lead to or facilitate the project's death. The paper shows the main causes that result in community disengagement and possible countermeasures. The following study analyzes the community relationships around the project while the authors analyze death behavior for SECOs.

Stack Overflow and Windows Phone SECO are used by Massanori et al. [2]. to examine the SECO death from the Developer Relations (DevRel) perspective. The following study analyzes Web SECOs while the authors analyze death in mobile systems.

Soupinski et al. [1] analyze the SECO death using development, engagement, and recycling metrics from the Stack Overflow and GitHub environments. The following study examines SECO death from the standpoint of social interaction.

### V. RESEARCH METHODOLOGY

The study uses the mining software repositories (MSR) guidelines by Hemmati et al. [4] to extract, transform, and load data (ETL). These guidelines show the best order or layers of activities and treatments to be applied to datasets.

As a pioneer study, we choose the analyses based on experimental software engineering concepts, with a focus on a specific context (web SECOS) and a unique objective (death web SECOS). Subsequently, we will not compare dead and alive SECOS.

The datasets, scripts, and analysis [6] are available to researchers and users that want to investigate deeper problems around death SECOS.

#### A. Goals and Research Questions

The main goal is to analyze the SECO death from a social perspective using the following indicators: (1) developers' recruitment and permanence; and (2) social collaboration, to understand this phenomenon in the software engineering landscape around Web SECOs.

- **RQ1.** Can the SECO maintain your contributors' activity before, during, and after your death? **Reason:** The number of developers present in a SECO shows your capacity to engage new workers and maintain old users to collaborate and evolve your software. **Metric:** Order.
- **RQ2.** Can the SECO maintain your contributors' relationship level before, during, and after your death? **Reason:** The number of relationships presents in a SECO shows the community's capacity to work together to solve problems and create new features rapidly. **Metric:** Average Degree.
- **RQ3.** Do SECO developers build strong relationships? **Reason:** The relationship strength shows the ability of users to create strong and fast communication channels with other developers resulting in the capacity to develop complex and important demands that require more than one person to work. **Metric:** Average Strength.

#### B. SECOs Selection Criteria

Starting the search, we explored Google for SECOs using the keywords "dead software platform" and "software platform death". To select a project from the results, we define that it has to present four principal characteristics.

- 1) The project has to be open source and accept contributions.
- 2) Official websites can be used to confirm the death.
- 3) The project has GitHub as a contribution environment.
- 4) The project is focused on the web.

After applying the criteria, the selected SECOs are listed in Table 1 with their launch, discontinuation, support start, and support end dates.

TABLE I  
LAUNCH, DISCONTINUATION, AND SUPPORT DATES

SECO	Launch	Discontinuation	Support Start	Support End
AngularJS	2009	2018/01	2018/07	2021/12
PhantomJS	2011/01	2016/01	2016/01	2018/03
Moment.js	2011	2020	2020	-

#### C. Data Extraction and Modelling

To extract data from GitHub, we used the Pydriller library [5] with Python scripts. We collect every registered issue or pull request (PL) from each repository and your comments.

We begin modeling the data with the first concept, which is the difference between relationships and interactions. For example, sending a message to an individual does not automatically establish a relationship between the sender and the receiver; they must also interact mutually (at least once as a sender and once as a receiver). Figure 1 illustrates this difference.

So, in formal terms, an interaction from user A to user B is user A's participation in an issue or pull request after user B. Subsequently, occurring also the inverse process (interaction from B to A) we can define a relationship between these users.

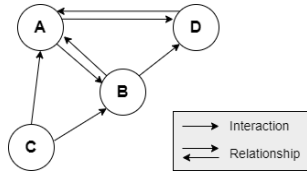


Fig. 1. Graphical Representation of Interactions and Relationships Between A, B, and C users

Another topic that has been discussed is the strength of the relationship. For example, when two coworkers work together to solve a problem, they form a strong bond, resulting in an open channel when one of them has a problem and needs help. These ties are especially useful in open-source communities where there is no financial incentive to implicate a coworker.

Therefore, we define a relationship's strength as the total of mutual interactions (minimum value between the interactions from user A to user B and from user B to user A).

#### D. Synthesis

To analyze community behavior from a social perspective, we need to define a group of metrics and concepts. Each of them is explained below.

- 1) **Order** is the number of users in the network.
- 2) **Size** is the number of relationships in the network.
- 3) **Average Degree** is the average number of user relationships.
- 4) **Average Strength** is the average strength of relationships in the network

#### VI. CAN SECO MAINTAIN YOUR CONTRIBUTORS' ACTIVITY BEFORE, DURING, AND AFTER YOUR DEATH?

When analyzing figure 2, we can see that in the first years of their lives, the projects attracted a large number of developers. This fact indicates the project's ability to engage users and communities to discover and collaborate on recent projects.

Another pertinent issue is the historical minimum of contributors following SECO's death. They are demonstrating a lack of support for users and other systems in migrating your requirements from SECO to other solutions.

The projects probably don't use methods for developer relations, which is a field of research about development communities that helps project owners use sustainable strategies to engage and support the synergy between their collaborators and goals. This makes it hard for developers to stay in the SECOS without engagement and rewards for their work, resulting in the instability of users inside the project.

#### VII. CAN SECO MAINTAIN YOUR CONTRIBUTORS' RELATIONSHIP LEVEL BEFORE, DURING, AND AFTER YOUR DEATH?

When analyzing Figure 3, we can see a similar behavior between the average degree and the number of developers. The average number increases during the first years of the project's life and then decreases to the historical minimum after the SECO's death. But, when we compared the proportional

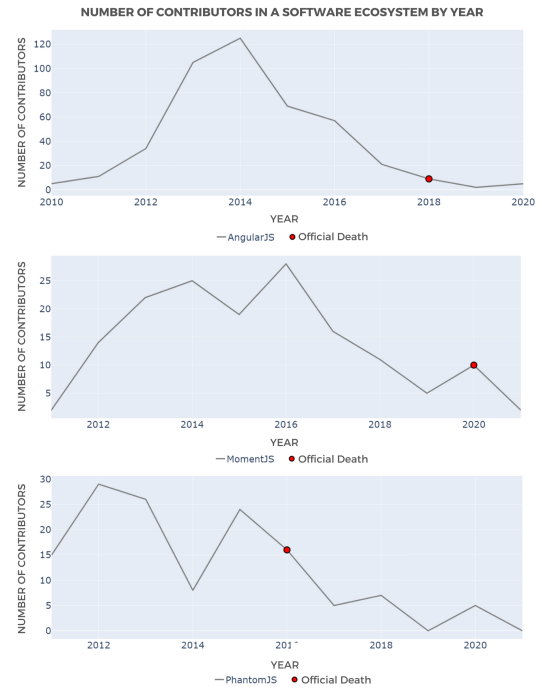


Fig. 2. Number of Contributors in a SECO by Year

increase between these metrics, we found a remarkable difference. The number of relationships does not follow the number of developers, demonstrating a new user's abandonment.

Another pattern is the low number of relationships, achieving at the historical maximum 2 relations by a user. This phenomenon serves as a warning to users and owners, particularly in open-source projects, because it strikes at one of its philosophical foundations, teamwork.

Complex and important demands that require more than one person to work on them may not be completed without a connected community, resulting in a static project. This occurs when issues are not investigated, new requirements are not attended to, pull requests are not verified, and more.

#### VIII. DO SECO DEVELOPERS BUILD STRONG RELATIONSHIPS?

The average strength has a particular behavior when compared with others. As the project evolves, the community in each SECO presents a specific movement. In AngularJS, the community can strengthen your relationships in a few years, but these relationships rot some years later. On the other hand, in the PhantomJS and MomentJS communities, the relationships gradually weaken.

Another relevant aspect and similarity between SECOS is the minimum average strength after project death. This indicates that the community cannot strengthen its ties without the engagement and management of project owners.

Furthermore, is notorious the ability of development communities to create strong relationships between their members, especially in the golden years of a project. So, researchers in future studies should analyze the impact of practices and



Fig. 3. Average Number of Relationships in a SECO by Contributor and Year

methods in developer relations on social aspects of project development.

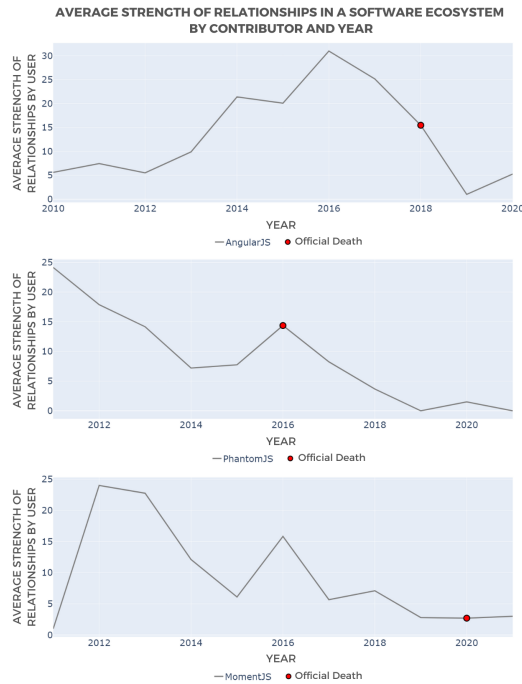


Fig. 4. Average Strength of Network SECOs

## IX. THREATS TO VALIDITY

On construct validity, the study's theoretical basis considered the fragilities shown in related works. The GitHub choice

is because of the strong presence of developers and your interactions around SECOS. The dataset was not aleatorily selected, but constructed with data around the studied SECOS.

To reduce the experimenter's expectations, the analysis followed the procedure indicated in studies around mining software repositories and realized conciliation meetings to analyze the proposed results.

This study is focused on web development tools and needs to be replicated in other areas, such as mobile and machine learning ecosystems.

## X. CONCLUSION

In this paper, we advance the study of social networks and community behavior around death SECOS focused on projects directed at web tools, such as AngularJS, PhantomJS, and Moment.js. From the search questions, we found some directions: **Users' recruitment and permanence:** Without management and engagement, communities get out of SECOS a few years after your entry, so managers and responsible organizations need to apply methods to control and organize the community around the platform. **Community relationships:** The community needs engagement to strengthen and maintain your relationships. Some programs, workshops, and events are useful methods to engage and encourage users to collaborate between them.

Benefits to industry and academia: analyze temporal series to identify SECO health through metrics about community activity and engagement. This method helps project owners create restoration or migration strategies in SECOS. Advances in knowledge about conditions that result in SECO deaths.

## XI. FUTURE WORK

In future research, living and long-lived SECOS should be analyzed to improve our understanding of useful and practical methods for organizing and engaging development communities in a SECO. Another exciting area is the SECOS health measurement, increasing the ability to predict the death event and preparing users and other software to migrate their requirements from a near-death SECO.

## REFERENCES

- [1] F. Soupinski, P. Arantes, I. Steinmacher, I. Wiese, H. Borges, B. Cafeo, and A. Fontão, "We are dying!" On Death Signals of Software Ecosystems. Proceedings of the XXXVI Brazilian Symposium on Software Engineering, pp. 363–369. October 2022.
- [2] D. Massanori, B. Cafeo, I. Wiese, and A. Fontão, Death of a Software Ecosystem: a Developer Relations (DevRel) perspective. Proceedings of the 34th Brazilian Symposium on Software Engineering, pp. 399–404. October 2020.
- [3] R. Evertse, A. Lencz, T. Šinik, J. Slinger and L. Soussi, Is Your Software Ecosystem in Danger? Preventing Ecosystem Death Through Lessons in Ecosystem Health. International Conference on Agile Software Development, pp. 96–105. October 2021.
- [4] H. Hemmati, S. Nadi, O. Baysal, O. Kononenko, W. Wang, R. Holmes and M. Godfrey. The msr cookbook: Mining a decade of research. 10th Working Conference on Mining Software Repositories (MSR), pp. 343–352. 2013.
- [5] PyDriller Documentation. Available at <https://pydriller.readthedocs.io/>. Last access at 2023/02/06.
- [6] Paper replication. Available at <https://github.com/pedrorodriguesarantes/-We-are-dying-Social-Networks-during-Software-Ecosystems-Deaths>. Last access at 2023/02/06.